ระบบปลดล็อคประตูอัตโนมัติ

Automatic door unlocking system

ณัฐธิดา ถนัดค้า รหัสนักศึกษา 62010270

Nutthida Thanadka Student ID. 62010270

พาณิภัค วงศ์สุทธิโกศล รหัสนักศึกษา 62010625

Panipak Wongsuttigosol Student ID. 62010625

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2565

ระบบปลดล็อคประตูอัตโนมัติ

Automatic door unlocking system

โดย

ณัฐธิดา ถนัดค้า

พาณิภัค วงศ์สุทธิโกศล

อาจารย์ที่ปรึกษา

ผศ.ดร. วีระ เพ็งจันทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2565

ปริญญานิพนธ์ปีการศึกษา 2565

ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
คณะ	วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง	ระบบปลดล็อคประตูอัตโนมัติ Automatic door unlocking system
ผู้จัดทำ	นางสาวณัฐธิดา ถนัดค้า รหัสนักศึกษา 62010270 นางสาวพาณิภัค วงศ์สุทธิโกศล รหัสนักศึกษา 62010625

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(ผศ.ดร. วีระ เพ็งจันทร์)
อาจารย์ที่ปรึกษา

หัวข้อโครงงาน	ระบบปลดล็อคประตูอัตโนมัติ	
นักศึกษา	นางสาวณัฐธิดา ถนัดค้า	รหัสนักศึกษา 62010270
	นางสาวพาณิภัค วงศ์สุทธิโกศล	รหัสนักศึกษา 62010625
ปริญญา	วิศวกรรมศาสตร์บัณฑิต	
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2565	
อาจารย์ที่ปรึกษาโครงงาน	ผศ.ดร. วีระ เพ็งจันทร์	

บทคัดย่อ

โครงงานฉบับนี้เป็นการศึกษา และออกแบบระบบปลดล็อคประตูอัตโนมัติ มีวัตถุประสงค์เพื่อศึกษาระบบ IoT ที่เกี่ยวกับ Smart Home และเพื่อทำการออกแบบระบบปลดล็อคประตูอัตโนมัติ โดยใช้ระบบเครือข่ายอินเตอร์เน็ต และควบคุมผ่านโทรศัพท์บนเว็บแอพพลิเคชัน เมื่อมีคนอยู่บริเวณประตูบ้านระบบนี้จะทำการแจ้งเตือนไปที่โทรศัพท์ของผู้ใช้งานผ่านฟังก์ชัน LINE Notify ในระบบจะประกอบไปด้วยเซนเซอร์ตรวจจับการเคลื่อนไหว ส่วนบอร์ด ESP32-CAM AI-Thinker ที่มีกล้อง OV2640 ใช้ในการถ่ายภาพแล้วส่งให้ผู้ใช้งานเห็นว่าคนที่อยู่หน้าบ้านเป็นใคร และสามารถสั่งการเปิดปิดประตูได้ผ่านเว็บแอพพลิเคชัน

Project Title	Automatic door unlocking system
Student	Miss. Natthida Thanadka Student ID 62010270
	Miss. Panipak Wongsuttigosol Student ID 62010625
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2022
Project Advisor	Asst. Prof. Dr. Weera Pengchan

ABSTRACT

This project is a study and design an automatic door unlocking system that has the purpose to study the IoT system related to smart home and to design an automatic door unlocking system using the internet network by control through the phone on the Web app. When person at the door this system will send a notification to the user's phone via LINE Notify. The system consists of the motion sensor, the ESP32-CAM AI-Thinker board with OV2640 camera is used to take a picture to the users that the person at the door is who. And the user can control the opening and closing of the door by pressing a button on the Web application.

กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาช่วยเหลือ แนะนำ ให้คำปรึกษา ตรวจสอบแก้ไขข้อบกพร่องต่าง ๆ ด้วยความเอาใจใส่อย่างยิ่งจากผศ.ดร.วีระ เพ็งจันทร์ และคณะอาจารย์จากภาควิชาวิศวกรรมอิเล็กทรอนิกส์ ผู้จัดทำกราบขอบพระคุณด้วยเป็นอย่างสูง ขอขอบคุณเพื่อน ๆ ทุกคนที่ให้ คำปรึกษา สุดท้ายขอขอบคุณครอบครัวที่ช่วยเหลือสนับสนุนทั้งด้านกำลังใจ และกำลังทรัพย์ด้วยดี ตลอดมา

ณัฐธิดา ถนัดค้า
พาณิภัค วงศ์สุทธิโกศล
นักศึกษาผู้จัดทำโครงงาน

สารบัญ

สารบัญรูปภาพ

สารบัญตาราง

ตารางที่ **หน้า**

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

การใช้ชีวิตของคนในสังคมเมืองปัจจุบันเป็นไปอย่างเร่งรีบ การคมนาคมบนท้องถนนทำได้อย่างยากลำบากเนื่องจากความหนาแน่นของจำนวนรถ รวมไปถึงสภาพถนนที่อาจไม่เอื้ออำนวยต่อการสัญจรไปมา เนื่องจากมีน้ำท่วมขัง เป็นหลุมเป็นบ่อ หรือมีการปิดถนนก่อสร้าง ซึ่งล้วนแล้วแต่เป็นสิ่งที่เลี่ยงไม่ได้ นอกจากนี้ในบางหมู่บ้านแม้จะมีการว่าจ้างผู้ดูแลความปลอดภัยแล้วก็ตามแต่ก็อาจจะมีเรื่องที่ไม่คาดคิดเกิดขึ้นได้เช่นกัน ปัญหาทั้งหมดนี้อาจทำให้เกิดความเหลื่อมล้ำทางด้านเวลาหากมีนัดใครไว้ที่บ้าน ทำให้คนที่รอพบเราอยู่ต้องรออยู่ภายนอกตัวบ้าน ประจวบกับในปัจจุบันเทคโนโลยีอินเตอร์เน็ตของสรรพสิ่ง หรือ ไอโอที (IoT) เข้ามามีบทบาทในชีวิตประจำวันเป็นอย่างมาก อุปกรณ์ทางอิเล็กทรอนิกส์มีการพัฒนาก้าวหน้าไปอย่างรวดเร็วเพื่อความสะดวกสบายในการใช้งานและทุกคนสามารถเข้าถึงได้

ดังนั้นโครงงานนี้จึงมีความสนใจที่จะศึกษา และต้องการสร้างระบบปลดล็อคประตูอัตโนมัติที่เจ้าของบ้านสามารถปลดล็อกประตูได้ทุกที่ทุกเวลาตามต้องการ พร้อมทั้งยังมีกล้องถ่ายภาพไว้สังเกตการณ์เมื่อมีสิ่งมีชีวิตเดินผ่านไปผ่านมาเสมือนเป็นระบบรักษาความปลอดภัยไปในตัว

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 สร้างระบบปลดล็อคประตูอัตโนมัติ

1.2.2 เพื่อศึกษาระบบ IoT ที่เกี่ยวกับ Smart Home

1.2.3 เพื่อศึกษาการใช้อุปกรณ์และเครื่องมือในการทำโครงงาน

1.2.4 เพื่อฝึกการคิดวิเคราะห์แก้ไขปัญหาที่จะเกิดขึ้นระหว่างการทำโครงงาน

1.3 ขอบเขตของโครงงาน

โครงงานนี้จัดทำขึ้นเพื่อศึกษาและสร้างระบบปลดล็อคประตูอัตโนมัติที่สามารถปลดล็อกประตูผ่านโทรศัพท์มือถือ ปลดล็อกประตูเมื่อป้อนรหัสผ่าน สามารถแจ้งเตือนและจับภาพเมื่อมีคนอยู่หน้าประตูได้ และสะดวกต่อการใช้งาน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถสร้างระบบปลดล็อคประตูอัตโนมัติได้

1.4.2 สามารถวิเคราะห์และแก้ปัญหาที่อาจจะเกิดขึ้นระหว่างการทำโครงงานได้

1.5 ระยะเวลาในการทำโครงงาน

ตารางที่ 1.1 ระยะเวลาในการทำโครงงาน

ตั้งแต่วันที่ 20 มกราคม ถึง 25 เมษายน พ.ศ.2566

<table>
<tr><th rowspan="2">ขั้นตอนการดำเนินงาน</th><th colspan="14">สัปดาห์ที่</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th></tr>
<tr><td>1.ทฤษฎี / ซื้ออุปกรณ์</td><td colspan="7">⟷</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2.ประกอบชุดทดลอง</td><td></td><td></td><td></td><td></td><td colspan="7">⟷</td><td></td><td></td><td></td></tr>
<tr><td>3.เก็บผลการทดลอง</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="6">⟷</td></tr>
<tr><td>4.สรุปผลการทดลอง</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="2">⟷</td></tr>
<tr><td>5.เขียนรายงานฉบับสมบูรณ์</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>⟷</td></tr>
</table>

บทที่ 2

หลักการและทฤษฎี

2.1 หลักการทำงานของระบบปลดล็อคประตูอัตโนมัติ

เซนเซอร์ตรวจจับการเคลื่อนไหวจะจับความเคลื่อนไหวของคนบนบริเวณประตูบ้าน แล้วจะใช้ บอร์ด ESP32-CAM AI-Thinker ที่มีกล้อง OV2640 ในการถ่ายภาพหลังจากนั้นจะส่งให้ผู้ใช้งานเห็นว่าคนที่อยู่หน้าบ้านเป็นใคร และผู้ใช้งานสามารถสั่งการเปิดปิดประตูได้ผ่านเว็บแอป

2.2 ทฤษฎีและส่วนประกอบที่เกี่ยวข้อง

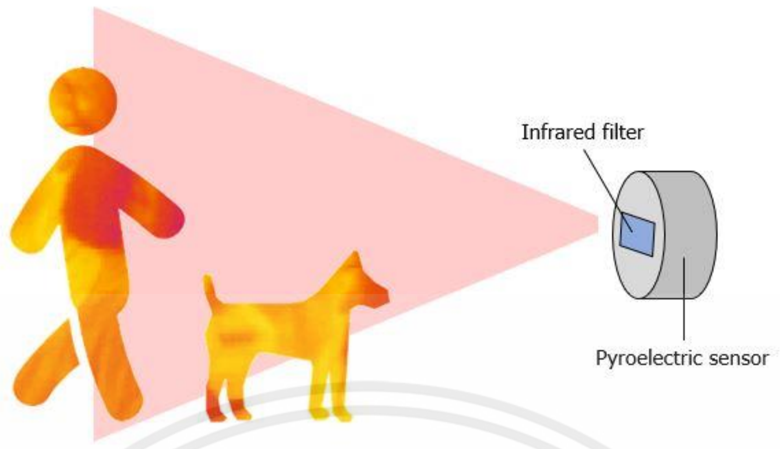
2.2.1 Motion Sensor

Motion Sensor หรือเซนเซอร์ตรวจจับการเคลื่อนไหว คือ อุปกรณ์สำหรับตรวจจับการเคลื่อนไหวด้วยสัญญาณชนิดหนึ่ง ก่อนจะแปลงเป็นสัญญาณทางไฟฟ้า เซนเซอร์ตรวจจับการเคลื่อนไหวมี 3 ประเภท ได้แก่

- Passive Infrared sensors (PIR) เป็นเซนเซอร์ตรวจจับคลื่นรังสีอินฟราเรดที่แผ่ออกมาจากวัตถุที่เป็นสิ่งมีชีวิต
- Ultrasonic เป็นเซ็นเซอร์ที่มีการปล่อยคลื่นอัลตร้าโซนิกออกมาและตรวจวัดการสะท้อนของคลื่นเมื่อวัตถุเคลื่อนที่ คำนวณเป็นระยะทางที่ได้จากการเดินทางของคลื่นและนำมาเทียบกับเวลา
- Microwave เป็นเซ็นเซอร์ที่มีการปล่อยคลื่นไมโครเวฟออกมาและตรวจวัดการสะท้อนของคลื่นเมื่อวัตถุเคลื่อนที่

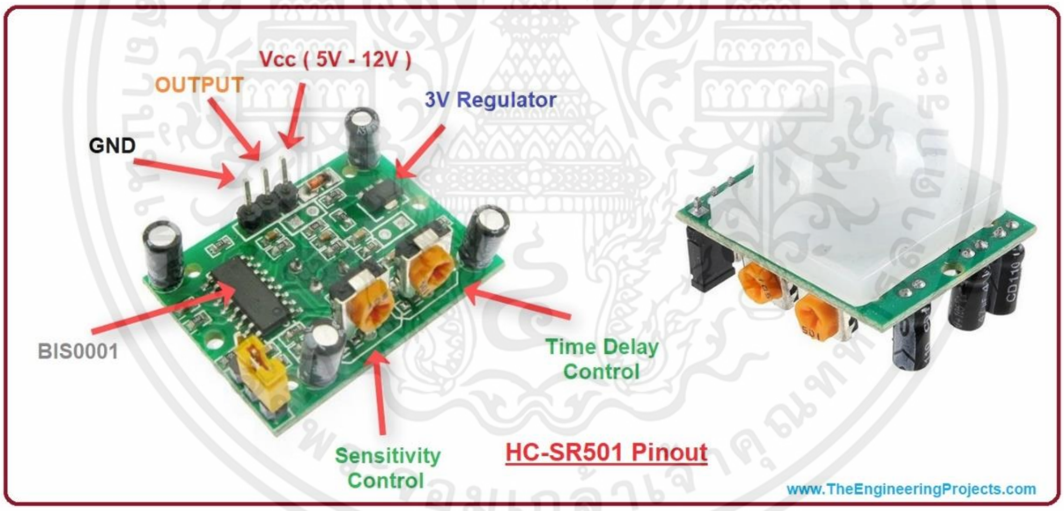
เซนเซอร์สำหรับโครงงานนี้จะเป็นเซนเซอร์ประเภท PIR รุ่น HC-SR501

Passive infrared sensor, Pyroelectric detector (PIR) เป็นเซนเซอร์ทำจากวัสดุประเภทไพโรอิเล็กทริก สามารถตรวจจับการเคลื่อนไหวด้วยรังสีอินฟราเรดที่แผ่ออกจากวัตถุเป้าหมาย (สิ่งมีชีวิต) ผ่านเลนส์ของตัวเซนเซอร์ แล้วแปลงอุณหภูมิของรังสีอินฟราเรดจากวัตถุเป้าหมายให้อยู่ในรูปของสัญญาณทางไฟฟ้าแบบดิจิตอลก่อนเข้าสู่วงจรอิเล็กทรอนิกส์เพื่อขยายสัญญาณก่อนจะส่งไปยังหน่วยประมวลผล



รูปที่ 2.1 วัตถุแผ่รังสีอินฟราเรดเข้าสู่เลนส์ของเซนเซอร์ตรวจจับการเคลื่อนไหว

[ที่มา https://www.jlhome.in.th/index.php?route=simple_blog/article/view&simple_blog_article_id=16]



รูปที่ 2.2 โมดูลเซนเซอร์ตรวจจับการเคลื่อนไหว HC-SR501

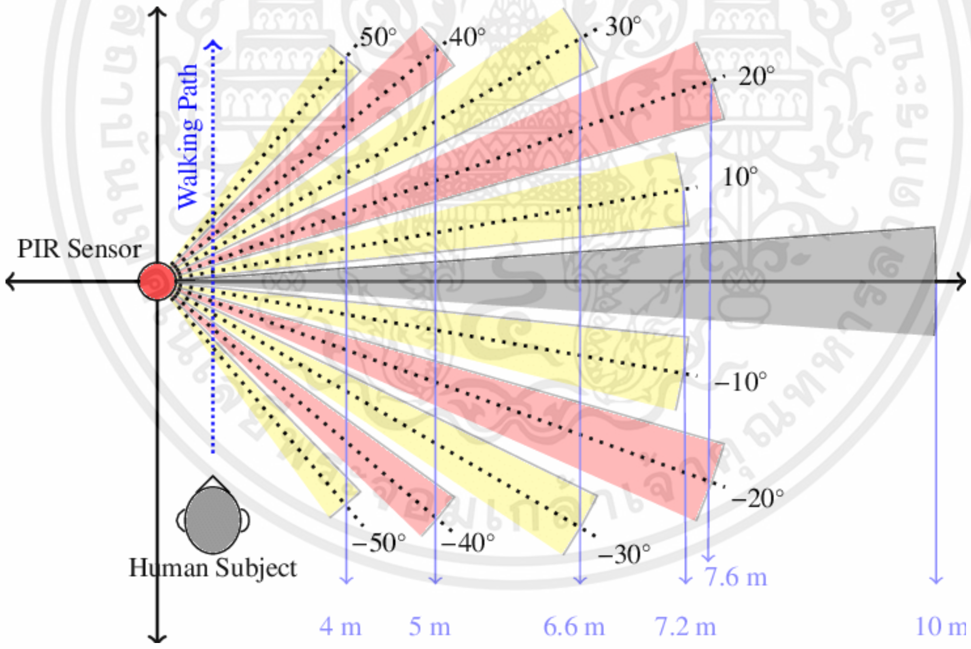
[ที่มา https://www.theengineeringprojects.com/2019/01/introduction-to-hc-sr501.html]

จากรูปที่ 2.2 ภายในโมดูล HC-SR501 ประกอบด้วย

- Vcc Pin สำหรับรับแรงดันขาเข้า 5-12V
- OUT Pin ส่งสัญญาณเอาท์พุตแบบดิจิตอลให้กับไมโครคอนโทรลเลอร์
- GND Pin สำหรับต่อกราวนด์

- Sensitivity control เป็นตัวต้านทานปรับค่าได้สำหรับการปรับระยะทางที่ PIR ตรวจจับได้ ระหว่าง 3-7 เมตร
- Time delay control เป็นตัวต้านทานปรับค่าได้สำหรับปรับการหน่วงเวลาของสัญญาณเอาท์พุต
- 3V Regulator เป็นวงจรแรงดันคงที่สำหรับลดแรงดันไฟฟ้ากระแสตรงให้อยู่ที่ 3VDC
- BIS0001 เป็นวงจรรวมที่รับสัญญาณอนาล็อกจาก Pyroelectric นำมาขยายสัญญาณ (Amplifier) และนำสัญญาณเข้าสู่การเปรียบเทียบ (Comparator) ก่อนออกไปเป็นสัญญาณเอาท์พุตแบบดิจิตอล
- Fresnel Lens หรือเลนส์เกลี่ยแสง ทำจากพลาสติกหรือกระจก ทำหน้าโฟกัสรังสีอินฟราเรดจากตำแหน่งต่างๆให้มาตกกระทบที่ Pyroelecteric

PIR sensors ประกอบด้วย Freshnel Lens ที่นอกจากจะช่วยรวมรังสีอินฟราเรดแล้วยังช่วยขยายขอบเขตของการตรวจจับวัตถุ (Field of View, FOV) เลนส์ประเภทนี้จะมีลักษณะเป็นรูปทรงกลมผ่าครึ่งทำให้ตรวจจับการเคลื่อนไหวของวัตถุ ในลักษณะที่ระยะทางไกลแต่มุมแคบ กับ ระยะทางใกล้แต่มุมกว้าง

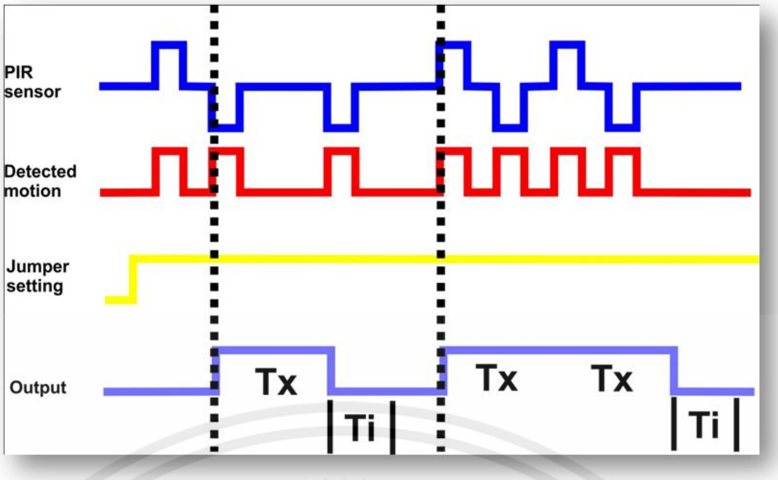


รูปที่ 2.3 ตัวอย่างการทดสอบ Field of View ของ PIR Sensors

[ที่มา https://www.researchgate.net/figure/Field-of-view-of-the-PIR-sensor-with-the-Fresnel-lens_fig3_325279130]

จากรูปที่ 2.3 เป็นตัวอย่างขอบเขตของการตรวจจับวัตถุของ Freshnel Lens ตั้งแต่มุม -50 ถึง 50 องศา ที่ระยะทาง 4 ถึง 10 เมตร ที่จะนำไปเป็นแบบทดสอบอ้างอิงสำหรับวัด Field of view ของโมดูลเซนเซอร์ที่จะใช้ในโครงงานนี้

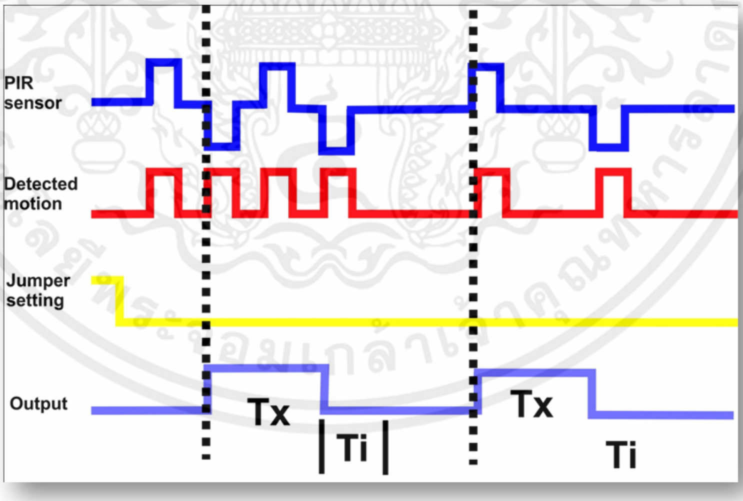
PIR Sensor Module รุ่น HC-SR501 สามารถตรวจจับก โหมดของสัญญาณเอาท์พุตได้ 2 แบบ คือ



รูปที่ 2.4 Output ของ PIR Sensor ในโหมด Retriggering

[ที่มา https://blog.thaieasyelec.com/getting-started-pir-motion-sensor/]

– H (Retriggering) เมื่อ PIR สามารถตรวจจับการเคลื่อนไหวได้จะส่งสัญญาณ Output ออกมาโดยจะส่ง Output ออกมาค้างสถานะเอาไว้จนกว่าจะไม่มีการเคลื่อนไหวจึงจะเปลี่ยนสถานะกลับไปยังสถานะปรกติ และจะเข้าสู่ช่วง Ti ซึ่งในช่วงนี้ตัว PIR จะไม่ตอบสนองการเคลื่อนไหวใดๆ



รูปที่ 2.5 Output ของ PIR Sensor ในโหมด Non – Retriggering

[ที่มา https://blog.thaieasyelec.com/getting-started-pir-motion-sensor/]

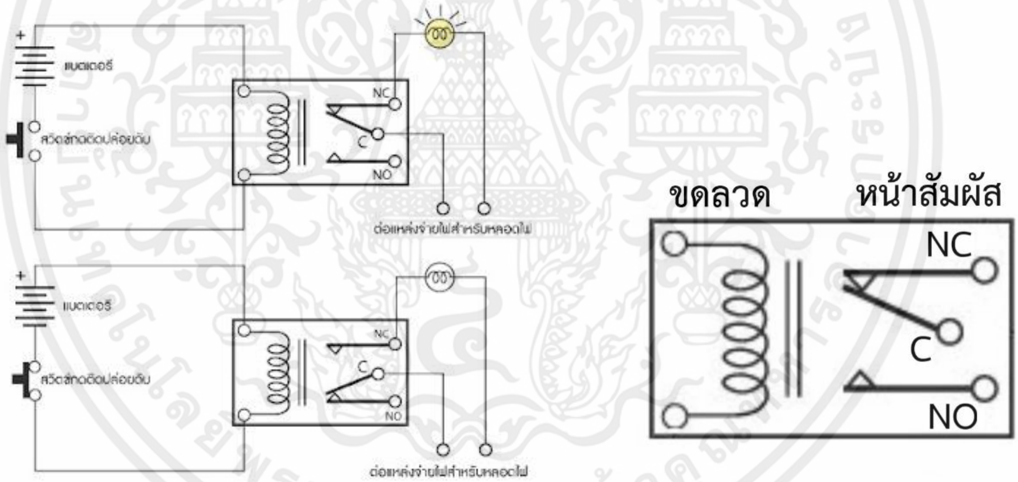
– L (Non – Retriggering) เมื่อตรวจจับการเคลื่อนไหวได้ PIR จะส่ง Output ออกมาและจะส่งออกมาค้างเอาไว้ในช่วงเวลาหนึ่ง (Tx) และจะเปลี่ยนสถานะกลับตามเดิม ถึงแม้ว่าในขณะนั้นจะยังคงมีการเคลื่อนไหวอยู่หรือไม่ และจะเข้าสู่สถานะ Ti คือจะไม่รับรู้การเคลื่อนไหวไปชั่วขณะหนึ่ง

2.2.2 Relay

รีเลย์ คืออุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก โดยจ่ายกระแสไฟฟ้าให้กับขดลวด เพื่อทำหน้าที่เปิดหรือปิดหน้าสัมผัสคล้ายกับอุปกรณ์อิเล็กทรกนิกส์ ใช้ในการควบคุมวงจรอิเล็กทรอนิกส์ต่างๆ เช่น การเปิดปิดอุปกรณ์ไฟฟ้า การสั่งให้อุปกรณ์มีการเคลื่อนไหวเป็นช่วงๆ

รีเลย์ประกอบไปด้วย

- ส่วนของขดลวด เริ่มทำงานเมื่อขดลวดได้รับแรงดันจากภายนอก จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะด้านในไปกระทุ้งให้แผ่นหน้าสัมผัสต่อกัน
- ส่วนของหน้าสัมผัส เสมือนเป็นสวิตช์ควบคุมการจ่ายกระแสไฟฟ้าให้กับวงจร
- จุดต่อ NO (Normally Open) เมื่อไม่มีการจ่ายกระแสไฟฟ้าให้กับขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน นำไปใช้กับวงจรที่ต้องการควบคุมการทำงานแบบเปิด-ปิดเอง
- จุดต่อ NC (Normally Close) เมื่อไม่มีการจ่ายกระแสไฟฟ้าให้กับขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน นำไปใช้กับวงจรที่ต้องการให้ทำงานตลอดเวลา
- จุดต่อ C (Common) คือจุดร่วมที่ต่อมาจากแหล่งจ่ายไฟ



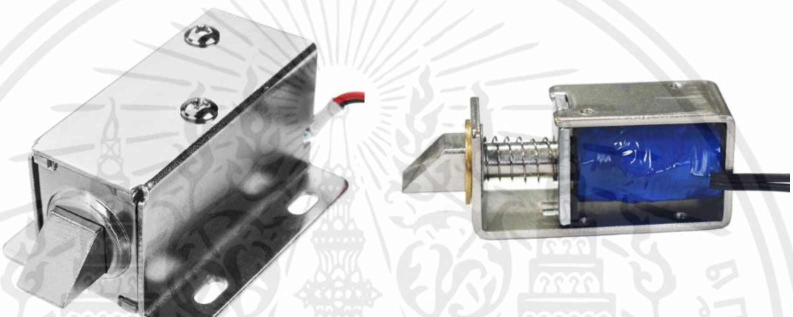
รูปที่ 2.6 โครงสร้างภายในของ Relay
[ที่มา https://www.iamall.co.th/content/4402/]

2.2.3 Solenoid Electromagnetic Door Lock

Solenoid Electromagnetic Door Lock กลอนประตูแม่เหล็กไฟฟ้าแบบโซลินอยด์ มีหลักการทำงานคือเมื่อจ่ายกระแสไฟฟ้าให้กับอุปกรณ์ กระแสไฟฟ้าจะไหลผ่านขดลวดโซลินอยด์จะกลายเป็นกลไกบังคับให้สลักกลอนหดเข้าไปเพื่อให้สามารถเปิดประตูได้ เมื่อไม่มีการจ่ายกระแสไฟฟ้า สลักกลอนก็จะกลับไปเป็นดังเดิม

โดยกลอนประตูแม่เหล็กไฟฟ้าที่นำมาใช้คือรุ่น LY-03 มีคุณสมบัติดังนี้

- ใช้แรงดัน 12VDC และกระแสขนาด 0.6A
- ใช้เวลาเปิดปิดประตูประมาณ 1 วินาที
- ทำงานในอุณหภูมิ -40 ถึง 50 องศาเซลเซียส



รูปที่ 2.7 Solenoid Electromagnetic Door Lock
[ที่มา https://www.arduino4.com/product/793]

2.2.4 ไมโครคอนโทรลเลอร์

ปัจจุบันการใช้งานอุปกรณ์อำนวยความสะดวกประเภทต่างๆ ทั้งในภาคครัวเรือนและภาคอุตสาหกรรม ได้มีการนำระบบคอมพิวเตอร์มาใช้ในการควบคุมการทำงาน อาทิ การใช้งานโทรศัพท์เคลื่อนที่และอุปกรณ์เทคโนโลยีสารสนเทศ เครื่องใช้ไฟฟ้าภายในบ้าน ยานพาหนะ รวมถึงเครื่องจักรกลการผลิตในอุตสาหกรรมประเภทต่างๆ เป็นต้น โดยระบบคอมพิวเตอร์ที่ใช้ในงานควบคุมดังกล่าวจะมีองค์ประกอบที่สำคัญ ดังนี้

1. หน่วยประมวลผลกลางหรือซีพียู (Central Processing Unit) เปรียบเสมือนสมองของคอมพิวเตอร์ ทำหน้าที่เป็นศูนย์กลางการประมวลผลและควบคุมระบบต่างๆ ของคอมพิวเตอร์

2. หน่วยความจำ

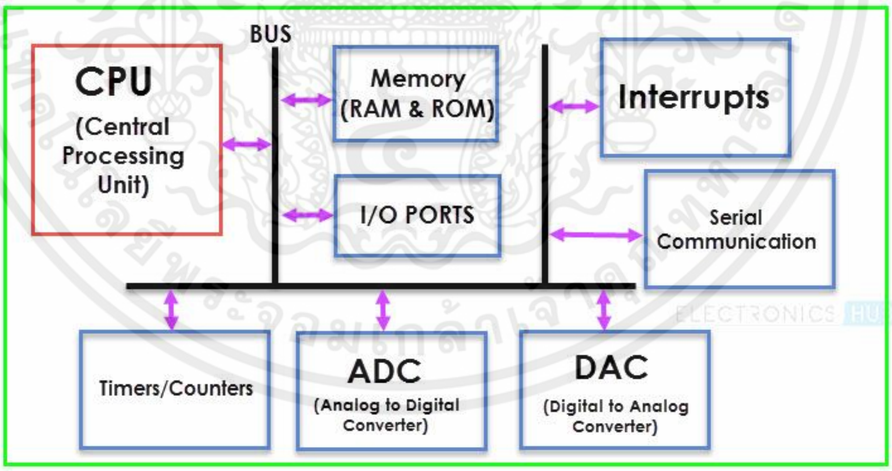
3. พอร์ตอินพุตและเอาต์พุต หมายถึงหน่วยรับหรือแสดงผลข้อมูล

รูปที่ 2.8 แผนผังของระบบคอมพิวเตอร์

[ที่มา https://www.nupress.grad.nu.ac.th/ไมโครคอนโทรลเลอร์/]

ไมโครคอนโทรลเลอร์ (อังกฤษ: Microcontroller มักย่อว่า μC, uC หรือ MCU) คือ อุปกรณ์ควบคุมขนาดเล็กซึ่งบรรจุความสามารถทีคล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาหน่วยความจำ หน่วยประมวลผลกลาง พอร์ตอินพุต/เอาท์พุตซึ่งเป็นส่วนประกอบหลักสาคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน

ไมโครคอนโทรลเลอร์ถ้าแปลความหมายแบบตรงตัวก็คือ ระบบคอนโทรลขนาดเล็กเรียกอีกอย่างหนึ่งคือเป็นระบบคอมพิวเตอร์ขนาดเล็กที่สามารถประยุกต์ใช้งานได้หลากหลายโดยผ่านการออกแบบวงจรให้เหมาะสมกับงานต่าง ๆ และสามารถโปรแกรมคำสั่งเพื่อควบคุมขา Input/Output เพื่อสั่งให้ไปควบคุมอุปกรณ์ต่าง ๆ ได้อีกด้วย ทั้งในทาง Digital และ Analog เช่น ระบบสัญญาณตอบรับอัตโนมัติ, ระบบบัตรคิว, ระบบตอกบัตร เป็นต้น และในปัจจุบันไมโครคอนโทรลเลอร์สามารถเชื่อมต่อกับระบบเครือข่ายอินเตอร์เน็ตเพื่อควบคุมอุปกรณ์ผ่านทางไกลได้อีกด้วย

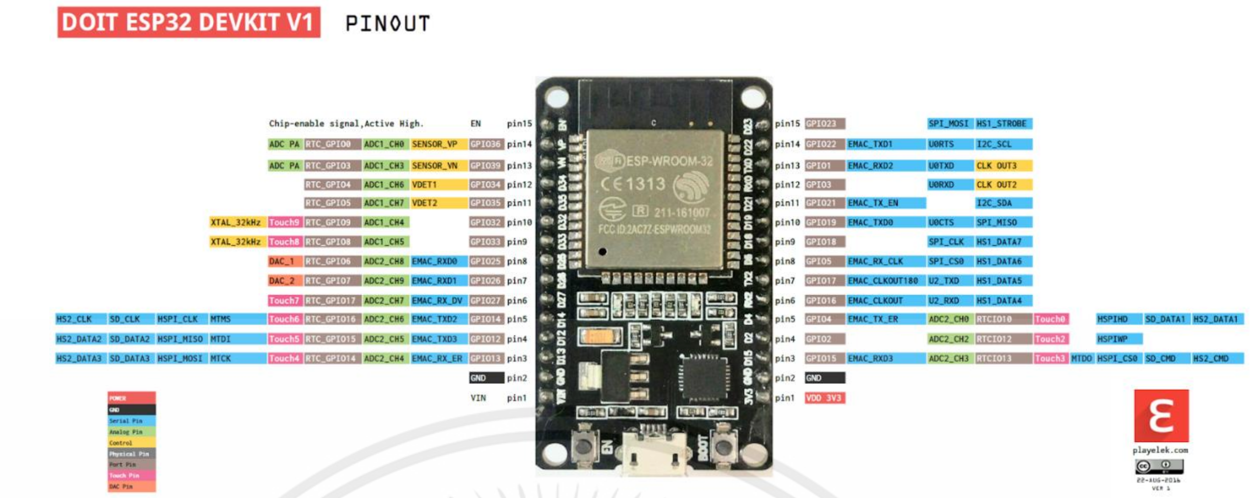


รูปที่ 2.9 โครงสร้างของไมโครคอนโทรเลอร์

[ที่มา https://www.electronicshub.org/microcontrollers-basics-structure-applications/]

สำหรับบอร์ดไมโครคอนโทรลเลอร์ที่จะใช้ในโครงงานนี้ จะพิจารณาจากบอร์ดที่สามารถเชื่อมต่อ WiFi ได้ มีราคาที่จับต้องได้และเขียนคำสั่งให้ทำงานตามต้องการได้สะดวก ได้แก่ ESP32 และ ESP32-CAM ซึ่งเป็นบอร์ดที่ถูกผลิตโดย Espressif Systems

2.2.5 ESP32 NodeMCU ESP-WROOM-32 (30 pins)



รูปที่ 2.10 DOIT ESP DEVKIT V1 Pinout

[ที่มา https://playelek.com/doit-esp32-devkit-v1/]

ESP32 เป็นชิปไมโครคอนโทรลเลอร์ที่มาพร้อม WiFi มาตรฐาน 802.11 b/g/n และบลูทูธเวอร์ชั่น 4.2 เป็นรุ่นต่อยอดความสำเร็จของ ESP8266 โดยในรุ่นนี้ได้ออกมาแก้ไขข้อเสียของ ESP8266 ทั้งหมด

โดย CPU ใช้สถาปัตยกรรม Tensilica LX6 จำนวน 2 คอร์ สัญญาณนาฬิกา 240MHz สามารถแยกการทำงานระหว่างโปรแกรมจัดการ WiFi และแอพพลิเคชั่นออกจากกันได้ ทำให้มีสเถียรภาพเพิ่มขึ้นมาก มีแรม 520KB มาในตัว นอกจากนี้ยังมี GPIO เพิ่มขึ้นมาก และมีช่อง ADC เพิ่มขึ้นเป็น 12 ช่อง จากเดิม ESP8266 มีเพียงช่องเดียว ใช้แรงดันไฟฟ้า 3.3V ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA ผลิตโดยบริษัท Espressif จากประเทศจีน โดยตัวไอซี ESP32 มีสเปคโดยละเอียด ดังนี้

- ซีพียูใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz
- มีแรมในตัว 512KB
- รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB
- มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi direct
- มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE
- ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V
- ทำงานได้ที่อุณหภูมิ -40°C ถึง 125°C

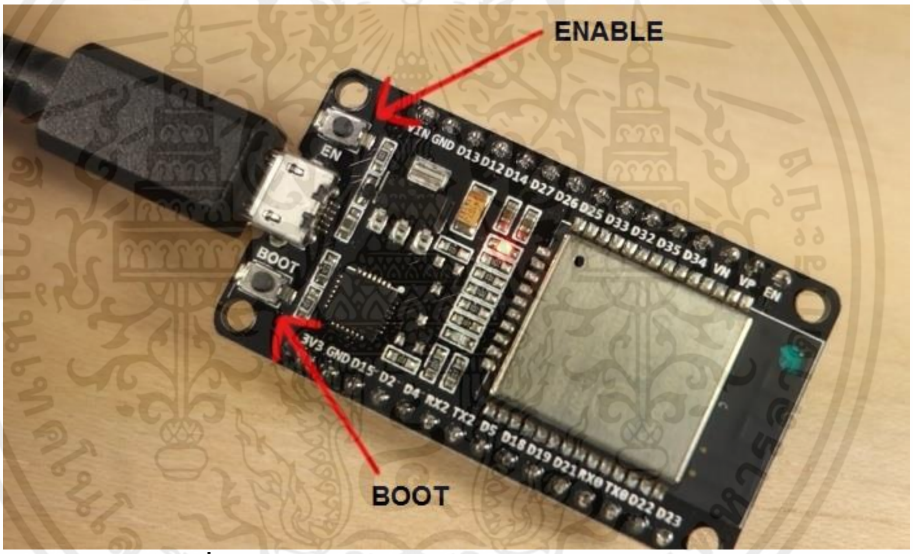
นอกจากนี้ ESP32 ยังมีเซ็นเซอร์ต่าง ๆ มาในตัวด้วย ดังนี้

- วงจรกรองสัญญาณรบกวนในวงจรขยายสัญญาณ
- เซ็นเซอร์แม่เหล็ก
- เซ็นเซอร์สัมผัส (Capacitive touch) รองรับ 10 ช่อง
- รองรับการเชื่อมต่อคลิสตอล 32.768kHz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะ

ขาใช้งานต่าง ๆ ของ ESP32 รองรับการเชื่อมต่อบัสต่าง ๆ ดังนี้

- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I2S จำนวน 2 ช่อง
- รองรับ PWM / Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card

ตัวบอร์ดสำเร็จรูปนี้มาพร้อมกับชิพ ESP-WROOM-32 มีวงจรแรงดันคงที่รับไฟขนาด 3.7-12V ปรับแรงดันให้คงที่ 3.3V เพื่อจ่ายไฟให้กับชิพ ESP32 มีภาคของวงจร USB TTL ใช้ชิพ CP2102 สำหรับติดต่อกับคอมพิวเตอร์ ตอนอัพโหลดโคด/Serial Monitor ผ่านทางสาย Micro USB ดังรูป



รูปที่ 2.11 การอัพโหลดโค้ด ESP32 ผ่าน Micro USB

[ที่มา https://www.cybertice.com/product/1149/esp32-nodemcu-esp-wroom-32-wi-fi-and-bluetooth-module-dual-core-consumption-cp2102]

2.2.6 ESP32-CAM

ESP32-CAM เป็นโมดูลกล้องขนาดเล็กที่ใช้พลังงานต่ำ ใช้ชิป ESP32-S มาพร้อมกับกล้อง OV2640 และมีช่องเสียบ SD Card ในตัว สามารถเชื่อมต่อ WiFi หรือ Bluetooth เพื่อการควบคุมอุปกรณ์จากระยะไกลได้ ในการประยุกต์ใช้ สามารถนำไปใช้กับระบบ IoT การถ่ายรูป กล้องวงจรปิด หรือระบบจดจำใบหน้า เป็นต้น

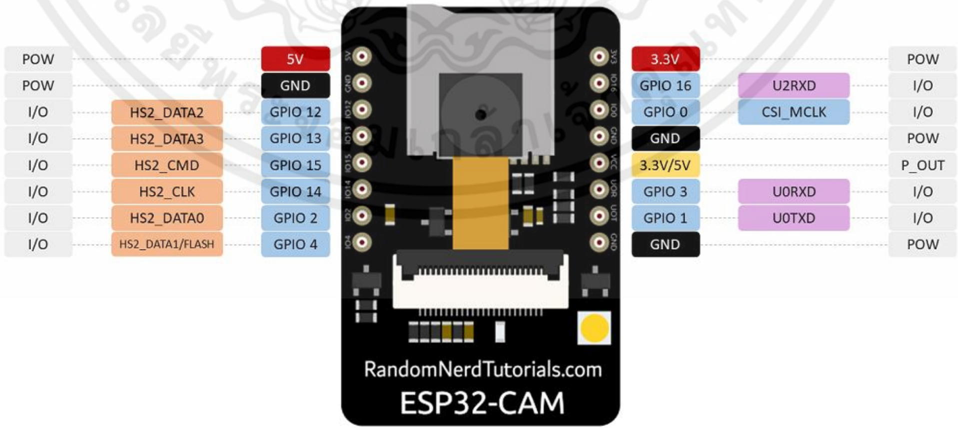
บอร์ด ESP32-CAM มาพร้อมกับเสาอากาศออนบอร์ด แต่ยังมีขั้วต่อ IPEX ช่วยให้คุณสามารถใช้เสาอากาศภายนอกเพื่อปรับปรุงช่วงการสื่อสาร Wi-Fi มาพร้อมกับปุ่มรีเซ็ตบนบอร์ดเพื่อรีสตาร์ท

บอร์ดและไฟ LED ในตัวที่ทำงานเป็นไฟแฟลช ไฟแฟลชมีประโยชน์ในการทำให้มีแสงแฟลชก่อนที่จะสตรีมวิดีโอ หรือถ่ายภาพ อย่างไรก็ตามไฟแฟลชจะใช้ GPIO ร่วมกับ GPIO ของการ์ด microSD ตัวใดตัวหนึ่ง ซึ่งหมายความว่าเมื่อใช้ฟังก์ชันที่เกี่ยวข้องกับการ์ด microSD ไฟ LED จะสว่างขึ้นเป็นครั้งคราวแม้ว่าจะไม่ต้องการก็ตาม

ข้อเสียที่สำคัญอย่างหนึ่งของ ESP32-CAM คือ ไม่มีอินเทอร์เฟส USB-to-UART ซึ่งไม่สามารถเชื่อม ESP32-CAM กับคอมพิวเตอร์ได้โดยตรงโดยใช้สาย USB ต้องใช้โปรแกรมเมอร์ FTDI อย่างไรก็ตามนี่เป็นหนึ่งในบอร์ดพัฒนา ESP32 ที่มีกล้องหลากหลาย และราคาถูกที่สุด

คุณสมบัติ (Features) ของบอร์ด ESP32-CAM AI-Thinker

1. The smallest 802.11b/g/n Wi-Fi BT SoC module
2. Low power 32-bit CPU, can also serve the application processor
3. Up to 160MHz clock speed, summary computing power up to 600 DMIPS
4. Built-in 520 KB SRAM, external 4MPSRAM
5. Supports UART/SPI/PWM/ADC/DAC
6. Support OV2640 and OV7670 cameras, built-in flash lamp
7. Support image WiFI upload
8. Support TF card
9. Supports multiple sleep modes
10. Embedded Lwip and FreeRTOS
11. Supports STA/AP/STA+AP operation mode
12. Support Smart Config/AirKiss technology
13. Support for serial port local and remote firmware upgrades (FOTA)



รูปที่ 2.12 ESP32-CAM AI-Thinker Pinout

[ที่มา https://i0.wp.com/randomnerdtutorials.com/wp-content/uploads/2020/03/ESP32-CAM-pinout-new.png?quality=100&strip=all&ssl=1]

ESP32-CAM AI-Thinker Pinout

1. Power Pins ใน ESP32-CAM มีพินกราวน์ (GND) ทั้งหมด 3 พิน และพินพาว์เวอร์ (Power) 2 พิน คือ 3.3V และ 5V ซึ่งสามารถใช้ไฟได้ทั้งพิน 3.3V และ 5V
2. Power output pin เป็นขาไฟเอาต์พุตที่สามารถจ่ายไฟ 3.3V หรือ 5V ให้กับอุปกร์อื่น ๆ ที่มาเชื่อมต่อกับ ESP32-CAM และไม่ควรใช้พินนี้จ่ายไฟให้กับ ESP32-CAM
3. Serial Pins คือ GPIO 1 (U0T/TX) และ 3 (U0R/RX) เพราะ ESP32-CAM ไม่มีโปรแกรมเมอร์ในตัวมันเอง ต้องใช้พินเหล่านี้เพื่อสื่อสารกับบอร์ด และอัพโหลดโค้ด ซึ่งสามารถอัพโหลดโค้ดได้โดยใช้ Arduino UNO R3 หรือ FTDI programmer ซึ่งสามารถใช้ GPIO 1 และ 3 เชื่อมต่ออุปกรณ์ต่อพ่วง ๆ เช่น เอาต์พุต หรือเซนเซอร์หลังจากการอัพโหลดโค้ดได้ อย่างไรก็ตามจะไม่สามารถเปิดดู Serial Monitor ที่ Arduino IDE ได้
4. GPIO 0 กำหนดว่า ESP32-CAM อยู่ในโหมดแฟลช (Flash Mode) หรือไม่ ซึ่ง GPIO 0 ได้ทำการต่อตัวต้านแบบพูลอัพ (Pull-up Resistor) 10 กิโลโอห์มไว้ภานในตัวบอร์ดแล้ว เมื่อ GPIO 0 เชื่อมต่อกับ GND ของ ESP32 จะเข้าสู่โหมดแฟลช และจึงจะสามารถอัพโหลดโค้ดได้ ซึ่งหลังจากการอัพโหลดโค้ดต้องหยุดการเชื่อมต่อระหว่าง GPIO 0 กับ GND ถึงจะสามารถทำงานได้ตามปกติ
5. MicroSD Card Connections พินต่อไปนี้ใช้เชื่อมต่อเพื่อเปิดใช้งานร่วมกับ MicroSD Card ถ้าหากไม่ได้ใช้งาน MicroSD Card สามารถใช้พินเหล่านี้เป็นพินอินพุต หรือเอาต์พุตได้ตาม ปกติ GPIO ทั้งหมดนี้เป็น RTC และรองรับ ADC และแสดงการเชื่อมต่อพินเพื่อเปิดใช้งาน ร่วมกับ MicroSD Card ดังตารางที่ 2.1

ตารางที่ 2.1 การเชื่อมต่อพินเพื่อเปิดใช้งานร่วมกับ MicroSD Card

MicroSD Card	Pin
CLK	GPIO 14
CMD	GPIO 15
DATA0	GPIO 2
DATA1 / flashlight	GPIO 4
DATA2	GPIO 12
DATA3	GPIO 13

[ที่มา https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout]

1. GPIO 4 – Flashlight ใน ESP32-CAM มี LED ในตัว ซึ่งสามารถทำงานเป็นแฟลชได้ขณะทำการถ่ายภาพ LED นั้นเชื่อมต่อภายในกับ GPIO 4 และใช้สำหรับการเชื่อมต่อเพื่อใช้งานกับ

MicroSD Card ด้วยหากมีปัญหาขณะใช้งานไฟแฟลชจะสว่างขึ้นเพื่อแจ้งเตือนว่าเกิดปัญหาระหว่างการใช้งาน MicroSD Card

2. GPIO 33 – Built-in Red LED ถัดจากปุ่ม Reset จะมีไฟ LED สีแดงอยู่บนบอร์ดด้วย LED นั้นเชื่อมต่อภายในกับ GPIO 33 สามารถใช้ LED นี้เพื่อระบุว่ามีบางอย่างเกิดขึ้น ตัวอย่างเช่น หากเชื่อมต่อ Wi-Fi แล้ว LED จะเป็นสีแดง หรือกลับกัน ซึ่งLED นั้นทำงานร่วมกับ Inverted logic จึงสามารถส่งสัญญาณ LOW เพื่อเปิด และสัญญาณ HIGH เพื่อปิด ซึ่งสามารถทำการทดลองโดยการอัพโหลดโค้ดลงบนบอร์ด ESP32-CAM ดังนี้

```
void setup() {
pinMode(33, OUTPUT);}
void loop() {
digitalWrite(33, LOW);}
```

3. Camera Connections การเชื่อมต่อกล้อง OV2640 ของ ESP32-CAM แสดงดังตารางที่ 2.2

ตารางที่ 2.2 การเชื่อมต่อกล้อง OV2640 ของ ESP32-CAM AI-Thinker

OV2640 CAMERA	Pin	Variable name in code
D0	GPIO 5	Y2_GPIO_NUM
D1	GPIO 18	Y3_GPIO_NUM
D2	GPIO 19	Y4_GPIO_NUM
D3	GPIO 21	Y5_GPIO_NUM
D4	GPIO 36	Y6_GPIO_NUM
D5	GPIO 39	Y7_GPIO_NUM
D6	GPIO 34	Y8_GPIO_NUM
D7	GPIO 35	Y9_GPIO_NUM
XCLK	GPIO 0	XCLK_GPIO_NUM
PCLK	GPIO 22	PCLK_GPIO_NUM
VSYNC	GPIO 25	VSYNC_GPIO_NUM
HREF	GPIO 23	HREF_GPIO_NUM
SDA	GPIO 26	SIOD_GPIO_NUM
SCL	GPIO 27	SIOC_GPIO_NUM
POWER PIN	GPIO 32	PWDN_GPIO_NUM

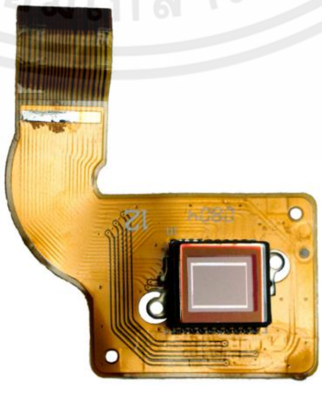
[ที่มา https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout]

สามารถกำหนดพินสำหรับ ESP32-CAM AI-Thinker บน Arduino IDE ควรเป็นดังนี้

```
#define PWDN_GPIO_NUM  32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM  0
#define SIOD_GPIO_NUM  26
#define SIOC_GPIO_NUM  27
#define Y9_GPIO_NUM    35
#define Y8_GPIO_NUM    34
#define Y7_GPIO_NUM    39
#define Y6_GPIO_NUM    36
#define Y5_GPIO_NUM    21
#define Y4_GPIO_NUM    19
#define Y3_GPIO_NUM    18
#define Y2_GPIO_NUM    5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM  23
#define PCLK_GPIO_NUM  22
```

2.2.7 Image sensor

เซนเซอร์รูปภาพ (image sensor) คืออุปกรณ์ที่แปลงภาพที่เห็นด้วยตาเป็นสัญญาณอิเล็กทรอนิกส์ โดยมากแล้วจะเป็นส่วนประกอบในกล้องดิจิตอล และอุปกรณ์ที่เกี่ยวกับภาพอื่นๆ เซนเซอร์ในยุคแรกๆ นั้นจะมีลักษณะเป็นกระบอกกล้องวีดิทัศน์ ต่อมาจึงพัฒนาเป็นอุปกรณ์ถ่ายเทประจุ หรือเซนเซอร์พิกเซลตอบสนอง (charge-coupled device - CCD) หรือแบบกึ่งตัวนำเมทัลอ็อกไซด์ควบเสริม (complementary metal-oxide-semiconductor - CMOS)



รูปที่ 2.13 เซนเซอร์รูปภาพ

[ที่มา https://en.wikipedia.org/wiki/Image_sensor]

1. ความลึกสีหรือ Color Depth หมายถึง จำนวนเฉดสีที่ Image Sensor สามารถถ่ายทอดออกมาได้ ยิ่งความลึกสีมาก จำนวนเฉดสีของภาพก็จะมากขึ้น หมายถึง เราจะได้ภาพที่มีคุณภาพดีขึ้นด้วย ความลึกสีจะบอกเป็นจำนวน Bit/สี หรือ Bit/3สี เช่น CCD ให้ภาพความลึกสี 12bit/สี ก็เท่ากับ 36 bit จำนวนเฉดสีที่ Image Sensor สามารถถ่ายทอดได้สามารถคำนวณได้โดยใช้สูตร

- จำนวนเฉดสี/สี = 2 ยกกำลัง Bit สี
- จำนวนเฉดสีทั้งหมด = จำนวนเฉดสี/สี ยกกำลัง 3

เช่น Image Sensor ให้ภาพ 8 bit/สี จะมีเฉดสี 28 = 256 สี จำนวนเฉดสีทั้งหมดเท่ากับ 2563 = 16.77 ล้านเฉดสี

Image Sensor ของกล้องดิจิตอลในปัจจุบันจะให้ความลึกสีที่ 8 bit/สี ถ้าเป็นกล้องที่คุณภาพดีจะอยู่ที่ 10 หรือ 12 bit/สี และถ้าเป็นกล้องระดับมืออาชีพจะอยู่ที่ 12-14 bit/สี ส่วนสแกนเนอร์คุณภาพสูงจะอยู่ที่ 16 bit/สี

- 12 bit/สี = 36 bit = 68,719, 476,736 หรือ 68,719 ล้านเฉดสี
- 16 bit/สี = 48 bit = 281,474, 976,710,656 หรือ 2.8 ล้านล้านเฉดสี

จะเห็นว่าจำนวน Bit สียิ่งมากจะยิ่งได้ภาพที่มีเฉดสีดีขึ้นเรื่อย ๆ ซึ่งกล้องระดับมืออาชีพจะเน้นเรื่องจำนวน Bit สีอย่างมาก ยิ่ง Bit สีมาก การไล่ระดับโทนสีในส่วนสว่างและส่วนมืดซึ่งเป็นปัญหาของกล้องดิจิตอลก็จะลดลงเรื่อย ๆ

2. Image Size หรือขนาดภาพ หมายถึงจำนวน Pixel ที่จะปรากฏบนภาพ ยิ่งจำนวน Pixel มากจะได้ภาพที่สามารถนำไปขยายใหญ่ได้มากขึ้นโดยไม่เกิดการแตก คล้ายกับฟิล์มเกรนหยาบกับเกรนละเอียด ขนาดภาพของ Image Sensor จะบอกเป็นจำนวน Effective Pixel เช่น กล้องมี Effective Pixel ขนาด 6.17 ล้านพิกเซล

การดูว่าจำนวน Pixel เท่าไรจะเพียงพอต่อการใช้งาน จะดูจากขนาดภาพที่ต้องการใช้งานเป็นหลัก เช่น ต้องการภาพไปใช้ส่ง E-Mail ซึ่งภาพจะมีขนาดประมาณ 4.87 แสนพิกเซล ใช้กล้องขนาด 1 ล้านพิกเซลก็เพียงพอ แต่ถ้าไปใช้งานขยายภาพขนาด 8.25x11.5 นิ้ว ควรมีความละเอียดประมาณ 8.5 ล้านพิกเซลจะได้ภาพคุณภาพสูงสุด เป็นต้น การใช้ Image Sensor ที่มีความละเอียดสูงเกินกว่าขนาดภาพที่ต้องการไม่เกิดประโยชน์ในการใช้งานใด ๆนอกจากจะต้องจ่ายค่ากล้องที่มีราคาแพงขึ้น ใช้แบตเตอรี่มากขึ้น เปลืองการ์ดเก็บข้อมูลมากขึ้น

จำนวน Pixel ของตามนุษย์ประมาณ 120 ล้านพิกเซล ฟิล์ม 35 มม.เกรนละเอียดมาก ๆ เช่น Fujichrome Provia 100F ขนาด 135มม.จะมีจำนวน Pixel อยู่ประมาณ 24 ล้านพิกเซล

3. Aspect Ratio หรือ สัดส่วนภาพ หรือสัดส่วนของภาพด้านกว้าง:ด้านยาว สัดส่วนตรงนี้มีความสำคัญกับการนำภาพไปใช้งาน เช่น ต้องการใช้อัดขยายภาพขนาด 4x6 นิ้ว เท่ากับภาพมีสัดส่วน 1:1.5 แต่ใช้กล้องดิจิตอลที่มีสัดส่วนกว้างยาว 1200x1600 พิกเซล หรือ 1:1.33 สัดส่วนกว้างยาวของภาพที่ต้องการและ Image Sensor ไม่เท่ากัน เมื่อนำภาพไปขยายจะได้ภาพไม่เต็มกระดาษ หรือเกิดการตัดส่วนภาพบนกระดาษไป กล้องดิจิตอลระดับมือสมัครเล่นจะมีสัดส่วนภาพอยู่ประมาณ 1:1.33 เพื่อให้เข้ากับจอมอนิเตอร์หรือ TV ส่วนกล้องดิจิตอลระดับมืออาชีพจะมีสัดส่วนประมาณ 1:1.5 ซึ่งเท่ากับฟิล์มขนาด 35 มม.

4. ความไวแสง หรือ Sensitivity ความไวแสงของ Image Sensor เป็นความไวแสงที่เทียบจากความไวแสงของฟิล์มในมาตรฐานของ ISO (International Standard Organization) ยิ่งความไวแสงสูงจะทำให้สามารถใช้ความเร็วชัตเตอร์สูงหรือช่องรับแสงแคบได้มากกว่า กล้องดิจิตอลส่วนใหญ่จะเริ่มความไวแสงที่ความไวแสงประมาณ ISO 100 แต่สามารถเลือกความไวแสงได้หลายค่าในกล้องตัวเดียว เช่น 100 , 200, 400, 800, 1600 ซึ่งไม่เหมือนฟิล์มที่จะไม่สามารถเปลี่ยนความไวแสงฟิล์มได้(ยกเว้นนำไปล้างเพิ่มหรือลดเวลาล้าง) และสามารถถ่ายภาพแต่ละภาพโดยใช้ความไวแสงที่แตกต่างกันได้ (ส่วนฟิล์มต้องตั้งความไวแสงค่าเดียวตลอดเวลา) ทำให้สะดวกในการใช้งานในสภาพแสงต่าง ๆกัน

การปรับตั้งความไวแสงสูงขึ้นในกล้องดิจิตอลจะเกิดสัญญาณรบกวน ทำให้ภาพมีคุณภาพลดลงไปบ้างเช่นเดียวกับการเพิ่มเวลาล้างของฟิล์มถ่ายภาพ

5. ขนาดของ Image Sensor หากเราใช้ตัว Image Sensor ขนาดใหญ่มีแนวโน้มจะให้คุณภาพที่ดีกว่า Image Sensor ขนาดเล็ก (จำนวน pixel เท่ากัน) เพราะจะมีขนาดของ Photosite ใหญ่กว่า ทำให้ไวต่อแสง มี Bit สีมากกว่า มีความคมชัดและรายละเอียดดีกว่า แต่ราคาจะแพงมากขึ้นตามขนาดของ Image Sensor ที่ใหญ่ขึ้น ตัวกล้องจะใหญ่ขึ้นตามด้วย จึงใช้เฉพาะกล้องระดับมืออาชีพเท่านั้นชนิดของ Image Sensor

ประเภทของเซนเซอร์รูปภาพ

Charge Coupled Device (CCD) เซนเซอร์แบบ CCD มีลักษณะเป็นอุปกรณ์อนาล็อก ประกอบไปด้วยหลอดโฟโต้ไดโอดทำด้วยซิลิคอน เมื่อแสงตกกระทบตัวชิปแล้วจะถูกเก็บไว้เป็นประจุอิเล็กทรอนิกส์ปริมาณน้อยๆ ในตัวเซนเซอร์ ซึ่งประจุเหล่านี้จะถูกแปลงไปเป็นแรงดันไฟฟ้าทีละหนึ่งพิกเซลขณะที่ถูกอ่านจากตัวชิป จากนั้นกระแสไฟในตัวกล้องจะแปลงแรงดันไฟฟ้านี้ไปเป็นข้อมูลดิจิทัลอีกทีหนึ่ง

Complementary Metal Oxide Semiconductor (CMOS) ส่วนแบบ CMOS มีลักษณะเป็นเซนเซอร์พิกเซลตอบสนอง ที่สร้างโดยกระบวนการประจุกึ่งตัวนำสำหรับ CMOS ซึ่งจะมีแผงวงจรเพิ่มขึ้นมาข้างเซนเซอร์ภาพเพื่อแปลงพลังงานแสงเป็นแรงดันไฟฟ้า จากนั้นแผงวงจรเสริมบนตัวชิปก็จะแปลงแรงดันไฟฟ้านั้นเป็นข้อมูลดิจิตอลได้ทันที

ข้อเปรียบเทียบเซนเซอร์แบบ CMOS VS CCD

- ชิปเซนเซอร์ CCD ผลิตด้วยเทคโนโลยี IC Wafer ดังนั้นต้นทุนการผลิตจึงสูงกว่าชิปเซนเซอร์ CMOS ซึ่งผลิตด้วยเทคโนโลยีสารกึ่งตัวนำ

- ภายในเซนเซอร์ CCD จะมีแต่วงจรส่วนที่ทำหน้าที่รับแสงเพียงอย่างเดียว ขณะที่ในเซนเซอร์ CMOS จะมีทั้งส่วนที่ทำหน้าที่รับแสงด้วยและแปลงสัญญาณด้วย ดังนั้นในขนาดพื้นที่เท่ากันเซนเซอร์ CCD จะมีความไวแสงสูงกว่า เพราะมีส่วนที่รับแสงได้มากกว่านั่นเอง แต่ในขณะเดียวกัน ก็จะเป็นผลให้เซนเซอร์ CCD มีความเร็วในการทำงานน้อยกว่า CMOS เพราะต้องเสียเวลาสงสัญญาณไปแปลงค่าอีกต่อหนึ่ง ขณะที่ CMOS จะทำงานเบ็ดเสร็จในตัวเองแล้ว

- CCD มีช่วงกว้างในการรับแสงกว้างกว่า คือสามารถรับแสงได้ตั้งแต่แสง Ultraviolet - UV ไปจนถึงแสง Infrared- IR

- คุณภาพภาพของชิปเซนเซอร์ CCD จะดีกว่า CMOS เพราะในพื้นที่เท่ากัน CCD รับแสงได้มากกว่า จึงมีความละเอียดมากกว่า ให้สีที่เหมือนจริงกว่า และให้เส้นแสงที่คมชัดมากกว่าด้วย

- CCD มีสัญญาณรบกวนภาพน้อยกว่า CMOS

- CCD ใช้พลังงานมากกว่า CMOS เนื่องจากต้องมีวงจรแปลงสัญญาณเพิ่มขึ้นมาอีก

จะเห็นได้ว่า CCD มีข้อได้เปรียบ CMOS ในเรื่องคุณภาพของภาพที่ได้ ขณะที่ CMOS ได้เปรียบในเรื่องความเร็วและการใช้พลังงานน้อยกว่า ซึ่งการที่ CCD ใช้พลังงานมากกว่าทำให้เกิดความร้อนสะสมสูง อันเป็นสาเหตุหนึ่งที่ส่งผลให้อายุการใช้งานสั้นลง แต่ในปัจจุบันเทคโนโลยีได้พัฒนาขึ้น ทำให้ CCD มีใช้พลังงานที่ต่ำลง ยืดอายุการใช้งานได้มากขึ้นอีกด้วย นอกจากนี้ CCD ยังมีข้อได้เปรียบ CMOS ในเรื่องความแข็งแรงของสัญญาณ ด้วยเหตุที่ CCD มีสัญญาณที่มารบกวนน้อยกว่า จึงสามารถส่งผ่านได้ในระยะที่ไกลกว่า โดยที่ยังรักษาคุณภาพของภาพเอาไว้ได้ นั่นคือสามารถเดินสายสัญญาณได้ไกลกว่า โดยที่ไม่ต้องอาศัยอุปกรณ์เสริมมาช่วย

โดยในโครงงานนี้ใช้เซนเซอร์รูปภาพรุ่น OV2640 ซึ่งเป็นโมดูลกล้องแบบ CMOS Camera ที่มาพร้อมกับบอร์ด ESP32-CAM ซึ่งสามารถถ่ายภาพได้สูงสุด 2 Megapixel ใช้การเชื่อมต่อแบบ Serial Camera Control BUS (SCCB) ซึ่งเป็นการสื่อสารเพื่อเชื่อมต่อโมดูลกล้องถ่ายรูปให้รับคำสั่งจากไมโครคอนโทรลเลอร์

รูปที่ 2.14 OV2640 Camera Module

[ที่มา https://www.robotsiam.com/product/211/ov2640-2-0-mp-interface-camera-module]

2.2.8 FTDI Programmer

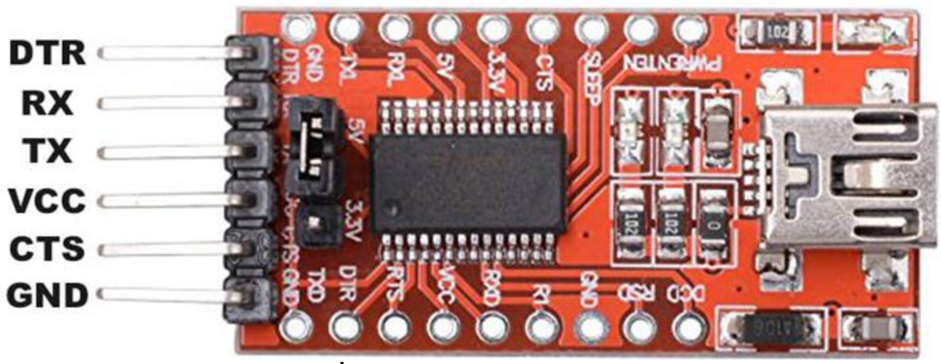
โมดูลตัวแปลงอนุกรม FTDI USB เป็น TTL เป็นบอร์ด UART (ตัวรับส่งแบบอะซิงโครนัส) ที่ใช้สำหรับการสื่อสารแบบอนุกรม TTL เป็นบอร์ดที่ใส่ชิป FTDI FT232R พร้อมอินเทอร์เฟซ USB สามารถใช้ 3.3 หรือ 5 V DC มีพอร์ต Tx หรือ Rx และพอร์ตอื่น ๆ ชิป FT232R ดั้งเดิมเป็นหนึ่งในชิป รุ่นใหม่ที่ผลิตโดย FTDI (Future Technology Devices International) นอกเหนือจากการเป็น USB เป็น UART แบบอนุกรมแล้ว ยังมี EEPROM ในตัว และเอาต์พุตตัวกำเนิดสัญญาณนาฬิกาเสริม ชิปยังรวม ฟังก์ชัน FTDIChip-ID (ทำให้แต่ละชิปมีตัวระบุที่ไม่ซ้ำกันเพื่อความปลอดภัย) และตัวต้านทานการ ยกเลิก USB บอร์ดโคลน (ที่มีชิปโคลน) มีแนวโน้มที่จะยกเว้นฟังก์ชัน ID ที่ไม่ซ้ำกัน และไม่รวม EEPROM ซึ่งหมายความว่าสัญญาณจะไม่สามารถกลับด้านได้ Clock ภายใน (6MHz, 12MHz, 24MHz และ 48MHz) สามารถนำออกจากอุปกรณ์ และใช้เพื่อขับเคลื่อนไมโครคอนโทรลเลอร์หรือตรรกะ ภายนอกได้ โมดูลนี้มีพอร์ต mini-B USB ซึ่งใช้เชื่อมต่อกับพอร์ต USB ของอุปกรณ์ที่เชื่อมต่ออยู่ (เช่นพีซี) อีกด้านหนึ่งโมดูลตัวแปลงอนุกรม FTDI USB เป็น TTL มีพินเบรคเอาต์ (Tx, Rx, Vc และ GND) ที่เชื่อมต่อกับพินที่เกี่ยวข้องของไมโครคอนโทรลเลอร์ การสื่อสารเกิดขึ้นโดยใช้ TTL (ลอจิกทรานซิสเตอร์ - ทรานซิสเตอร์) มีจัมเปอร์ DC 3.3 หรือ 5V และจ่าย 500 mA ผ่านขา Vcc

คุณสมบัติ (Features) ของ FTDI Programmer

1. ใช้ชิพ FT232R (FTDI)
2. 6 pin headers and 18 through-hole connectors. DTR, RX, TX, VCC, CTS, GND and PWR, TEN, SLEEP, CTS, 3.3V, 5V, RXL, TXL, GND x 2, TXD, DTR, RTS, VCC, RXD, R1, RSD, DCD (all marked)
3. Working Voltage 5 V DC
4. Level output 3.3 – 5 V DC (jumper)
5. Interfaces Serial/UART
6. Operating current 500mA
7. Module size 36 x 18 mm
8. Pin size male, 5 x 2.54 mm
9. Ports Micro-B USB

Pinout ของ FTDI Programmer

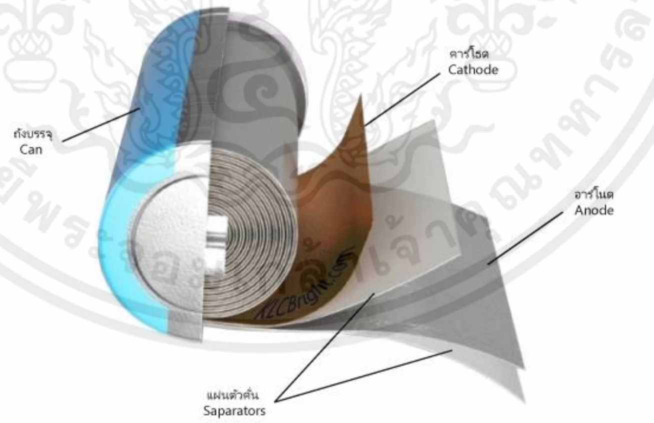
โมดูลตัวแปลงอนุกรม FTDI USB เป็น TTL มีหมุดส่วนหัวหกตัวตัวผู้ 5 x 2.54 มม. และขั้วต่อ 18 รู พินและมีการทำเครื่องหมายหมุด หรือขั้วต่อทั้งหมดไว้อย่างชัดเจน พินที่ใช้กันทั่วไป (RX, TX, VCC และ GND) จะเชื่อมต่อกับพินที่เกี่ยวข้องของไมโครคอนโทรลเลอร์ (RX -> TX และ TX -> RX) พิน DTR เชื่อมต่อกับพินรีเซ็ตของไมโครคอนโทรลเลอร์ซึ่งจะทริกเกอร์สัญญาณรีเซ็ตอัตโนมัติหลังจาก อัพโหลดโค้ด



รูปที่ 2.15 FTDI Programmer

2.2.9 แบตเตอรี่ลิเธี่ยม (Lithium Battery)

แบตเตอรี่ลิเธี่ยม (Lithium Battery) หรือ เรียกแบบเต็มว่า แบตเตอรี่ลิเธี่ยมไออน (Lithium Ion Battery) นั้นเรียกชื่อตามการกักเก็บพลังงานในลิเธี่ยมไออน (Li ion) โดยการสร้างศักย์ไฟฟ้าให้เกิดขึ้นระหว่างขั้วบวกและขั้วลบของแบตเตอรี่และคั่นด้วยฉนวนที่เรียกว่า ตัวคั่น (Saparator) ตัวคั่นสามารถเป็นได้ทั้งฉนวนและนำไอออน เมื่อตอนขณะชาร์จไฟ ลิเธี่ยมไออนจะเคลื่อนที่จากขั้วบวกไปยังขั้วลบผ่านตัวคั่น และเมื่อตอนคายประจุ ลิเธี่ยมไออนจะเคลื่อนที่ในทิศตรงกันข้าม การเคลื่อนที่ของประจุลิเธี่ยมไออนนี้เองทำให้เกิดความต่างศักย์ไฟฟ้าหรือที่เรียกว่าโวลท์ (Voltage) เกิดขึ้น แต่เมื่อไม่มีการใช้งานแบตเตอรี่ตัวคั่นจะทำหน้าที่เป็นฉนวนกั้นไม่ให้ไอออนเคลื่อนที่ข้ามไปมา และเมื่อมีการเอาแบตเตอรี่ไปต่อเข้ากับอุปกรณ์เพื่อใช้งาน ประจุลิเธี่ยมไออนหรืออิเล็กตรอนที่ถูกกั้นอยู่ด้วยตัวคั่นนั้นจะถูกดันให้วิ่งผ่านตัวกั้นได้



รูปที่ 2.16 แบตเตอรี่ลิเธี่ยม (Lithium Battery)

ชนิดของแบตเตอรี่ลิเธี่ยม (Types of Lithium Battery)

ประเภทหรือชนิดของแบตเตอรี่ลิเธี่ยมนั้น ส่วนใหญ่เรียกชนิดตามองค์ประกอบของคาร์โธด แต่ก็มีที่เรียกตามองค์ประกอบของอาร์โนดด้วยเช่นกัน ดูจากภายนอกเราไม่สามารถจำแนกได้ ดังนั้นเราจึงควรตรวจสอบให้แน่ชัดและศึกษารายละเอียดให้แน่ใจก่อนเลือกซื้อและนำไปใช้งาน ทั้งนี้เพื่อความปลอดภัย

ประสิทธิภาพสูงสุดและมีคุณสมบัติตรงตามแต่ละประเภทการใช้งานของอุปกรณ์หรือเครื่องมือนั้นๆ ชนิดของแบตเตอรี่ลิเธี่ยมมี 6 ชนิดหลักๆ ได้แก่

1.แบตเตอรี่ลิเธี่ยมฟอสเฟต (Lithium Ion Phosphate : LFP)

แบตเตอรี่ลิเธี่ยมฟอสเฟต หรือ LiFePO4 หรือ LFP หรือเราอาจจะรู้จักกันในชื่อ LiPO หรือที่บ้านเรานิยมเรียกกันว่า แบตฯลิโป เป็นแบตเตอรี่ลิเธี่ยมที่มีใช้ทั่วไปและใช้อยู่มากที่สุด ใช้ฟอสเฟตเป็นคาร์โธด ใช้กราไฟต์เป็นอาร์โนด แบต LiPO มีรอบอายุการใช้งานที่นาน ไม่ค่อยมีปัญหาด้านความร้อน มีประสิทธิภาพด้านเคมีไฟฟ้าที่ดี แบตฯ LFP หรือ LiPO 1 cell มีความต่างศักย์หรือแรงดันไฟฟ้าประมาณ 3.2V ถ้าต้องการนำไปใช้งานเป็นระบบ 12V ต้องเอามาต่ออนุกรมกัน 4 cell จะได้ 12.8V จึงเป็นเหตุผลว่า ทำไมแบต LiPO จึงนำมาใช้อย่างแพร่หลายที่สุด รวมถึงไฟถนนโซล่าเซลล์ เพราะสามารถใช้ 4cell แล้วนำมาใช้แทนแบตแบบตะกั่วหรือแบบ deep cycle ได้

ข้อเด่นของแบตเตอรี่ลิเธี่ยมฟอสเฟต

- อายุการใช้งานนาน ปกติแล้วแบตเตอรี่ลิเธี่ยมฟอสเฟตมีอายุการใช้งาน 2000 cycles หรือมากกว่า
- deep of discharge สูง ปกติแล้วแบตเตอรี่ลิเธี่ยมฟอสเฟตมี deep of discharge สูงประมาณ 80% สูงถึง 100% สำหรับผู้ผลิตบางรายโดยไม่ได้ทำให้แบตเตอรี่เสียหาย
- ปลอดภัยและทนต่ออุณหภูมิได้สูง

ข้อด้อยของแบตเตอรี่ลิเธี่ยมฟอสเฟต

- แบตเตอรี่ลิเธี่ยมฟอสเฟตมีค่าพลังงานจำเพาะไม่สูง
- ประสิทธิภาพจะลดลงเมื่อใช้งานในที่ที่มีอุณภูมิต่ำ
- ไม่ค่อยเหมาะใช้งานที่มีการใช้งานแบบกระแสกระชากสูงๆ (high cranking applications) เป็นเพราะด้วยคุณสมบัติ ข้อด้อย2ข้างข้างบนจึงทำให้แบตเตอรี่ลิเธี่ยมฟอสเฟตไม่ค่อยเหมาะใช้งานที่มีการใช้งานแบบกระแสกระชากสูงๆ เช่น การสตาร์ทเครื่องยนต์

2.แบตเตอรี่ลิเธี่ยมโคบอลต์ออกไซด์ (Lithium Cobalt Oxide : LCO)

แบตเตอรี่ลิเธี่ยมโคบอลต์ออกไซด์ หรือ LCO หรือที่บ้านเรานิยมเรียกว่า แบตลิเธี่ยมไอออน (ซึ่งไม่ถูกต้อง เพราะว่าแบตลิเธี่ยมไอออนเป็นคำที่กว้าง ที่กำลังกล่าวถึงทั้งหมดล้วนเป็น ลิเธี่ยมไอออน) มีค่าพลังงานจำเพาะ (specific power) สูงแต่มีกำลังงานจำเพาะ(specific power)ต่ำ กล่าวคือสามารถจ่ายไฟได้นานแต่จ่ายไฟให้กับอุปกรณ์ที่กินไฟสูงได้ไม่ดี จึงเหมาะและมีใช้ในงาน อุปกรณ์อิเล็กทรอนิกส์ เช่น โทรศัพท์มือถือ แทบเลต โน้ตบุ๊ค กล้องถ่ายรูป ที่ต้องการกำลังน้อยๆ แต่อยู่ได้นานๆ โดยที่ แบตเตอรี่ลิเธี่ยมโคบอลต์ออกไซด์ 1cell มีแรงดันไฟฟ้าประมาณ 3.7V

ข้อเด่นของแบตเตอรี่ลิเธี่ยมโคบอลต์ออกไซด์ LCO

- ข้อดีของแบตเตอรี่ลิเธี่ยมโคบอลต์ออกไซด์ คือมีค่าพลังงานจำเพาะสูงสามารถจ่ายไฟให้กับอุปกรณ์ที่กินไฟต่ำได้นาน

ข้อด้อยของแบตเตอรี่ลิเธี่ยมโคบอลต์ออกไซด์ LCO

– มีอายุการใช้งานสั้น ประมาณ 500-1000cycles
– วัตถุดิบ โคบอลต์ มีราคาค่อนข้างสูง
– ทนความร้อนได้ต่ำจึงมีปัญหาเรื่องความความปลอดภัยอยู่พอสมควร
– ไม่เหมาะใช้งานกับอุปกรณ์ที่กินไฟสูง

3.แบตเตอรี่ลิเธี่ยมแมงกานิสออกไซด์ (Lithium Manganese Oxide : LMO)

แบตเตอรี่ลิเธี่ยมแมงกานิสออกไซด์ หรือ LMO ใช้ส่วนผสมของธาตุลิเธี่ยมแมงกานีสออกไซด์ เป็นคาร์โธด ซึ่งองค์ประกอบของธาตุสามชนิดนี้ เมื่อรวมกันจะสร้างให้เกิดโครงสร้างแบบ 3 มิติ ทำให้การไหลของอิออนดีขึ้น มีความต้านทานภายในต่ำลง เพิ่มกระแสได้มากขึ้นและมีค่าการทนต่อความร้อน ได้มากขึ้นด้วย การค้นพบแบตฯ LMO (ประมาณ ค.ศ.1981) ถือเป็นความสำเร็จก้าวสำคัญในวงการ แบตเตอรี่ ด้วยความที่มีความต้านทานภายในเซลล์ต่ำจึงสามารถชาร์จแบบเร็ว (fast charge) และจ่ายกระแสสูง (high dis-charge current) ได้ แบตฯ LMO สามารถที่จ่ายกระแสได้ถึง 20-30A โดยที่ค่าความร้อนไม่ขึ้นสูงมากได้ แต่แบตฯ LMO ก็มีค่าการเก็บพลังงานได้น้อย และอายุการใช้งานที่ ไม่ได้ยาวมากนัก แบตเตอรี่ชนิด LMO มีใช้งานในอุปกรณ์ประเภทเครื่องมือไร้สาย (power tools) อุปกรณ์ทางการแพทย์ รถไฟฟ้าและรถไฮบริดบางรุ่น

ข้อเด่นของแบตเตอรี่ลิเธี่ยมแมงกานิสออกไซด์
– ชาร์จเร็ว
– มีค่าพลังงานจำเพาะสูงกว่าแบตฯ LCO
– ขนาดเล็กกว่า LCO
– จ่ายกระแสได้สูง

ข้อด้อยของแบตเตอรี่ลิเธี่ยมแมงกานิสออกไซด์
– มีอายุการใช้งานได้สั้น เพียง 300-700cycles เท่านั้น และสั้นที่สุดในบรรดาแบตเตอรี่ลิเธี่ยม ทั้งหมด

4.แบตเตอรี่ลิเธี่ยมนิเคิลแมงกานิสโคบอลต์ออกไซด์ (Lithium Nikel Manganese Cobalt Oxide:NMC)

แบตเตอรี่ลิเธี่ยมนิเคิลแมงกานิสโคบอลต์ออกไซด์ หรือ NMC ด้วยการเพิ่มนิเคิล Ni ซึ่งมีความจุพลังงานจำเพาะ (Specific Capacity;Ah/kg) สูง เข้าไปในส่วนประกอบของคาร์โธด ทำให้แบตฯ NMC มีค่าพลังงานจำเพาะสูงขึ้น ผลที่ได้คือได้เซลล์ที่มีความต่างศักดิ์หรือแรงดันไฟฟ้าสูงถึงประมาณ 4V/cell เดิมทีเดียวแบตฯ ชนิด NMC ไม่ได้มีส่วนผสมของแมงกานิส ต่อมานักวิจัยจึงได้เพิ่มแมงกานิสเข้าไป เพื่อเพิ่มความสเถียร ผลก็คือแบตฯ NMC มีทั้งความสเถียรและใช้งานได้ในงานแรงดันสูง (high voltage applications) การปรับเปลี่ยนส่วนผสมระหว่าง นิเคิล แมงกานีส และโคบอลต์ ทำให้เกิดชนิดย่อย และคุณสมบัติที่แตกต่างกันและยังคงเป็นจุดขายของผู้ผลิตแต่ละราย เช่น รุ่น NMC111 (ส่วนผสมเท่ากัน) รุ่น NMC442 รุ่น NMC622 หรือ รุ่น NM811 ในปัจจุบันแบตฯ NMC มีความต้องการสูงในอุตสาหกรรม รถยนต์ EV เพราะด้วยคุณสมบัติ มีค่าพลังงานสูง จ่ายกระแสได้สูง มีสเถียรภาพ และทนต่ออุณหภูมิได้ดีมาก มีใช้ในรถยนต์ไฟฟ้าหลายรุ่น เช่น Tesla Model S, Nissan Leaf, Chevloret Volt, BMW i3

ข้อเด่นของแบตเตอรี่ลิเธี่ยมนิเคิลแมงกานิสโคบอลต์ออกไซด์ NMC
– มีค่าพลังงานจำเพาะสูง

– สามารถจ่ายกระแสได้มาก

– ชาร์จเร็ว

– มีอายุการใช้งานนาน

– ปลอดภัย

ข้อด้อยของแบตเตอรี่ลิเธี่ยมนิเคิลแมงกานิสโคบอลต์ออกไซด์ NMC

– มีแรงดันไฟฟ้าต่อเซลล์ต่ำกว่า LCO เล็กน้อย

5.แบตเตอรี่ลิเธี่ยมนิเคิลโคบอลต์อลูมิเนียมออกไซด์ (Lithium Nikel Cobalt Aluminum Oxide : NCA)

แบตเตอรี่ลิเธี่ยมนิเคิลโคบอลต์อลูมิเนียมออกไซด์ หรือ NCA มีคุณสมบัติคล้ายกับ NMC คือเก็บไฟได้มาก และจ่ายไฟได้นาน แต่มีข้อเสียคือด้านความปลอดภัยยังไม่ค่อยสูงต้องมีระบบตรวจเช็คความปลอดภัยอยู่ตลอดเวลาเมื่อนำมาใช้ในรถ EV จนถึงปัจจุบันมี Tesla เพียงเจ้าเดียวที่ใช้แบตฯ NCA มีในรุ่น Model 3 และ Model S รุ่นแรก (2012)

ข้อดีแบตเตอรี่ลิเธี่ยมนิเคิลโคบอลต์อลูมิเนียมออกไซด์ NCA

– มีค่าพลังงานจำเพาะสูง

– สามารถจ่ายกระแสได้มาก

– ชาร์จเร็ว

– มีอายุการใช้งานนาน

ข้อด้อยของแบตเตอรี่ลิเธี่ยมนิเคิลโคบอลต์อลูมิเนียมออกไซด์ NCA

– ความปลอดภัยต่ำกว่าชนิดอื่น

– ราคาสูง

6.แบตลิเธี่ยมไททาเนต (Lithium Titanate : LTO)

แบตเตอรี่ลิเธี่ยมไททาเนต หรือ LTO ที่เปลี่ยนอาร์โนดจากกราไฟต์เป็นลิเธี่ยมไททาเนต ส่วนคาร์โธด มีทั้งเป็น LMO หรือ LMC ผลคือได้แบตฯที่ทั้ง ชาร์จเร็วกว่าแบตฯชนิดอื่นๆ อายุการใช้งานนาน และมีความปลอดภัยขึ้น แบตลิเธี่ยมไททาเนต (LTO) มีใช้ใน รถยนต์ EV เวอร์ชั่นที่มีใช้ภายในประเทศ ญี่ปุ่นเอง เช่น รถยนต์ไฟฟ้า Mitsubishi i-MiEV , Honda Fit EV , Tosa Electric Concept Bus นอกจากนี้ยังมีใช้ในเครื่องมือทางการทหาร ไฟถนนโซล่าเซลล์ รถ EV สถานีชาร์จรถ EV UPS เครื่องมือสื่อสาร รวมไปถึงอุปกรณ์ทางอวกาศ

ข้อดีแบตเตอรี่ลิเธี่ยมไททาเนต LTO

– ชาร์จเร็ว

– มีช่วงอุณหภูมิใช้งานที่กว้างมาก

– มีอายุการใช้งานนาน

– มีความปลอดภัยและความสเถียรสูงที่สุด

ข้อด้อยของแบตเตอรี่ลิเธี่ยมไททาเนต LTO

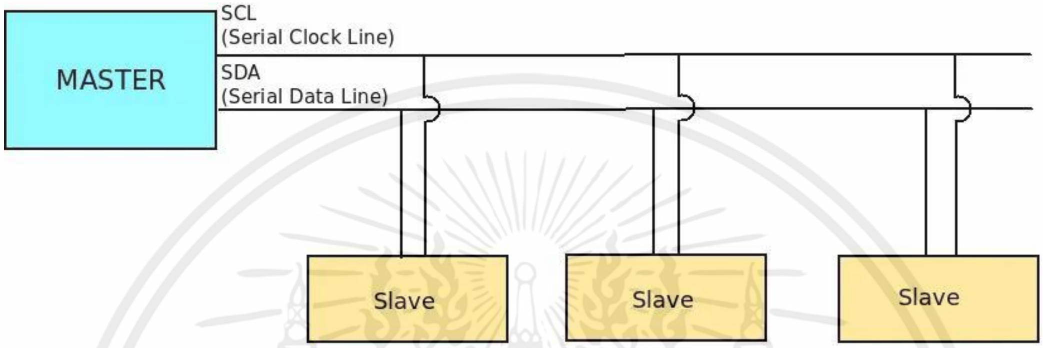
– มีค่าพลังงานจำเพาะต่ำ แบตฯจึงมีขนาดใหญ่

– ราคาสูงมาก

2.2.10 โปรโตคอลการสื่อสาร

การสื่อสารอนุกรมแบบ I2C : I2C ย่อมาจาก Inter-Integrated Circuit เป็นการสื่อสารข้อมูลความเร็วต่ำแบบซิงโครนัส ข้อดีของการสื่อสารเช่นนี้คือสามารถรับส่งข้อมูลได้หลายอุปกรณ์ในบัสเดียวกัน และใช้สายสัญญาณเพียง 2 เส้นในการรับส่ง-ข้อมูล ทำให้สามารถลดสายสัญญาณที่ใช้ในการเชื่อมต่ออุปกรณ์ลงมาก โดยสายสัญญาณทั้ง 2 เส้นแบ่งเป็น

- SDA ส่งสัญญาณนาฬิกา เพื่อบอกว่าข้อมูลที่จะรับ/ส่ง จะทำที่ความถี่เท่าใด
- SCL ใช้ในการส่งสัญญาณข้อมูลเพียงอย่างเดียว



รูปที่ 2.17 แผนผังการสื่อสารอนุกรมแบบ **I2C**

[ที่มา https://medium.com/icreativesystems/i²c-คืออะไร-ทำไมชื่อไม่คุ้นเลย-b79b9c18aaa6]

การใช้งานการสื่อสารอนุกรมแบบ I2C

สำหรับการใช้งานการสื่อสารอนุกรมแบบ I2C จะใช้รูปแบบการสื่อสารที่เป็นมาตรฐานของการสื่อสารอนุกรมแบบ I2C มีขั้นตอนการเขียนข้อมูลด้วยการสื่อสารอนุกรมแบบ I2C และการอ่านข้อมูลด้วยการสื่อสารอนุกรมแบบ I2C ดังนี้

การเขียนข้อมูลด้วยการสื่อสารอนุกรมแบบ I2C มีขั้นตอนดังนี้

1. Master Device ส่งสัญญาณ Start ไปยัง Slave Device
2. Master Device เขียนไบต์ควบคุม (Control Byte) ไปยัง Slave Device (ไบต์ควบคุม คือ Address ของ Slave ขนาด 7 บิต และคำสั่งเขียนข้อมูล (0) จำนวน 1 บิต)
3. Master Device เขียนข้อมูลไปยัง Slave Device
4. Master Device ส่งสัญญาณ Stop ไปยัง Slave Device

การอ่านข้อมูลด้วยการสื่อสารอนุกรมแบบ I2C มีขั้นตอนดังนี้

1. Master Device ส่งสัญญาณ Start ไปยัง Slave Device
2. Master Device ส่งไบต์ควบคุม ไปยัง Slave Device (ไบต์ควบคุม คือ Address ของ Slave Device ขนาด 7 บิตและตามด้วยคำสั่งเขียนข้อมูล (0) จำนวน 1 บิต)
3. Master Device ส่งสัญญาณ Stop ไปยัง Slave Device
4. Master Device ส่งสัญญาณ Start ไปยัง Slave Device
5. Master Device ส่งไบต์ควบคุมไปยัง Slave Device (ไบต์ควบคุม คือ Address ของ Slave Device ขนาด 7 บิตและตามด้วยคำสั่งอ่านข้อมูล (1) จำนวน 1 บิต)

6. Master Device อ่านข้อมูลจาก Slave Device
7. Master Device ส่งสัญญาณ Stop ไปยัง Slave Device

รูปที่ รูปแบบสื่อสารอนุกรมแบบ I2C

สำหรับการพัฒนาโปรแกรมการสื่อสารอนุกรมแบบ I2C บนแพลตฟอร์ม Arduino จะใช้งานผ่านไลบรารี่ Wire ที่ถูกติดตั้งมาบนโปรแกรม Arduino IDE เรียบร้อยแล้ว เมื่อผู้ใช้ต้องการพัฒนาโปรแกรมบนโปรแกรม Arduino IDE จึงสามารถเรียกใช้งานผ่านการ #include <Wire.h>

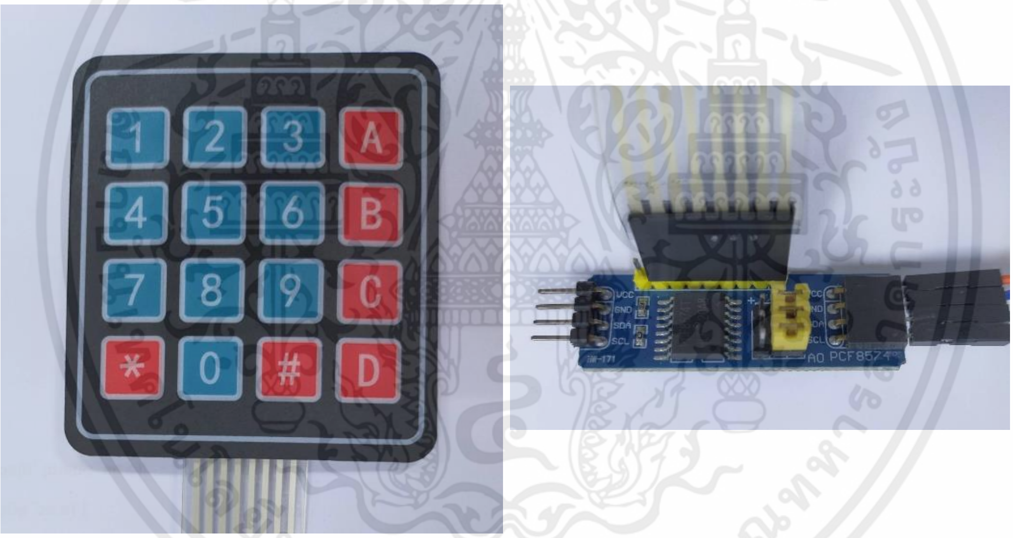
ฟังก์ชันการใช้งานการสื่อสารอนุกรมแบบ I2C ด้วยไลบรารี่ Wire จากนั้นกำหนด Wire.begin()

ฟังก์ชันสำหรับเริ่มต้นใช้งานการสื่อสารอนุกรมแบบ I2C

ในโครงโครงงานนี้จะมีการสื่อสารระหว่างอุปกรณ์และไมโครคอนโทรลเลอร์เกิดขึ้นที่ Keypad และจอ LCD

2.2.11 I2C Matrix 4x4 Keypad

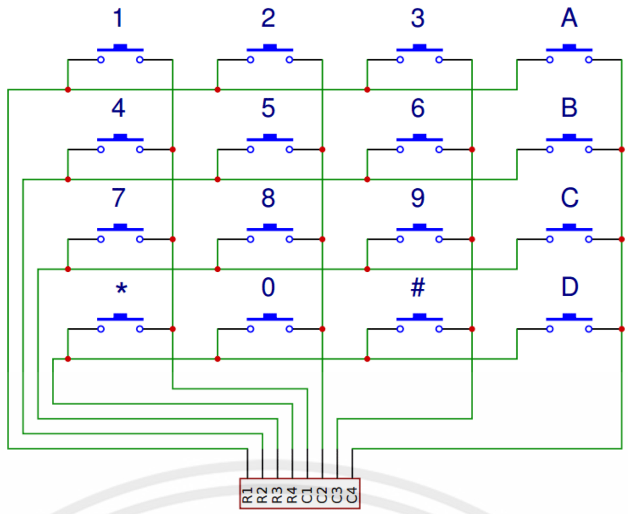
เป็นโมดูล Matrix Keypad 4x4 แบบ Membrane พร้อมชิพ PCF8574 โมดูลขยายพอร์ต Digital Input / Output สั่งงานผ่านทาง I2C เพื่อลดขาเชื่อมต่อระหว่าง Microcontroller กับ Keypad



รูปที่ 2.18 Matrix Keypad 4x4 และ โมดูลขยายพอร์ต PCF8574

Keypad เป็นหนึ่งในอุปกรณ์อินพุตที่นิยมใช้ในงานด้านระบบสมองกลฝังตัวมากที่สุด เราสามารถพบเห็นการใช้งาน Keypad ได้จากตู้อัตโนมัติต่าง ๆ เช่น ตู้เติมเงินมือถือ ตู้ซื้อขนมปัง ตู้ ATM ซึ่งแต่ละตู้นำมาใช้เพื่อให้ผู้ใช้บริการกรอกข้อมูลที่เป็นตัวเลขเข้าไป เช่น หมายเลขโทรศัพท์ จำนวนเงิน หมายเลขสินค้า หมายเลขบัญชี รหัสผ่านบัตร ATM แม้เราจะพบเห็นการใช้งาน Keypad ได้ทั่วไป

Keypad มักจะบอกขนาดเป็นจำนวนแถว x จำนวนคอลัมน์ เช่น มี 1 แถว 4 คอลัมน์ มักจะเขียน 1x4 หรือมี 4 แถว 3 คอลัมน์ มักจะเขียน 4x3 และ 4 แถว 4 คอลัมน์ จะเขียนเป็น 4x4



รูปที่ 2.19 โครงสร้างภายในของ Keypad

[ที่มา https://cz.lnwfile.com/_/cz/_raw/ya/4z/5k.png]

โครงสร้างภายในของ Keypad เป็นปุ่มกดที่นำแต่ละปุ่มมาต่อกันเป็นวงจรแบบเมตริกซ์ จะแบ่งชุดสายไฟได้ 2 กลุ่ม คือกลุ่มด้านคอลั่มน์ และกลุ่มด้านแถว โดยสายด้านกลุ่มคอลั่มน์ หรือแถว จะต้องมีการต่อตัวต้านทาน Pull-up หรือ Pull-down อยู่ตลอดเวลา การตรวจหาว่าปุ่มใดถูกกดอยู่จะใช้วิธีที่เรียกว่า การสแกน

การสแกน สามารถทำได้หลายวิธี ดังนี้

วิธีที่ 1 สแกนด้านเดียว คือการกำหนดเลือกแถว หรือคอลั่มน์ เป็นอินพุต ส่วนอีกด้านเป็นเอาต์พุต สมมุติให้ด้านแถวเป็นอินพุต มีตัวต้านทานต่อ Pull-down อยู่ และด้านคอลั่มน์เป็นเอาต์พุต

- เริ่มกำหนดให้คอลั่มน์ที่ 1 มีสถานะเป็นลอจิก 1
- วนรอบหาแถวที่มีลอจิกเป็น 1 สมมุติว่าเจอแถวที่ 2 มีลอจิกเป็น 1
- นำค่าของคอลั่มน์ (คือ 1) และค่าของแถว (คือ 2) ไป map กับค่าในตัวแปรอาเรย์ 2 มิติ สมมุติให้ค่าในตำแหน่งของอาเรย์ 2 มิติ [(1-1)][(2-1)] มีค่าเท่ากับ '2' แสดงว่าปุ่มเลข 2 ถูกกดอยู่
- การสแกนแบบนี้ จะต้องทำทั้งหมด จำนวนแถว * จำนวนคอลั่มน์ (สมมุต Keypad ขนาด 4x4 จะต้องสแกนทั้งหมด 16 ครั้ง) เพราะต้องสแกนหาทุก ๆ ปุ่มของ Keypad ซึ่งวิธีนี้อาจจะง่าย แต่จะค่อนข้างจะเปลืองรอบการทำงานของไมโครคอนโทรลเลอร์มาก

วิธีที่ 2 สแกนสองด้าน ในการเขียนโปรแกรมมักใช้วิธีการกำหนดค่าลงไปในรีจิสเตอร์โดยตรง โดยจะประหยัดขั้นตอนการทำงานลงได้มาก แต่จะมีตัวต้านทานต่อ Pull-up หรือ Pull-down อยู่ที่สายทุกเส้น ขั้นตอนการสแกนวิธีนี้ มีดังนี้

- สั่งให้ด้านแถวมีสถานะเป็นอินพุต

- สั่งให้ด้านคอลั่มน์(ทุกคอลั่มน์)มีสถานะเป็นเอาต์พุต และปล่อยลอจิก 1 ออกไป
- อ่านค่าที่ได้ทางด้านแถวมาทั้งหมด สมมุติให้อ่านออกมาแล้วพบว่าแถวที่ 2 มีสถานะเป็นลอจิก 1 แสดงว่าหนึ่งในปุ่มของแถวที่ 2 ถูกกดอยู่
- สั่งให้ด้านคอลั่มน์มีสถานะเป็นอินพุต
- สั่งให้ด้านแถวมีสถานะเป็นเอาต์พุต และปล่อยลอจิก 1 ออกไป
- อ่านค่าที่ได้ทางด้านคอลั่มน์มาทั้งหมด สมมุติให้อ่านออกมาแล้วพบว่าคอลั่มน์ที่ 3 มีสถานะเป็นลอจิก 1 แสดงว่าหนึ่งในปุ่มของคอลั่มน์ที่ 3 ถูกกดอยู่
- นำค่าที่ได้จากการสแกน 2 ครั้ง มา map ในตัวแปรอาเรย์ 2 มิติ สมมุตให้ค่าค่าในอาเรย์ 2 มิติ [(2-1)][(3-1)] มีค่าเท่ากับ '6' แสดงว่าปุ่มเลข 6 ถูกกดอยู่
- จะเห็นว่าการสแกนสองด้านจะมีขั้นตอนที่มากกว่า และเปลืองตัวต้านทานมากกว่า แต่วิธีนี้จะสแกนเพียงสองครั้ง แล้วได้ตำแหน่งของปุ่มที่ถูกกดทันที

Keypad ที่นิยมใช้งานกันในปัจจุบันมักจะมีขนาด 4x4 และ 4x3 ซึ่งต้องใช้สายจำนวน 7 - 8 เส้น ในการต่อเข้ากับไมโครคอนโทรลเลอร์ ซึ่งอาจไม่มีปัญหากับไมโครคอนโทรลเลอร์ 8 บิต มากนัก แต่มักจะเป็นปัญหากับไมโครคอนโทรลเลอร์ 32 บิต เนื่องจากจำนวนขา GPIO มีน้อย ไมโครคอนโทรลเลอร์ 32 บิตหลาย ๆ เบอร์ แต่ละขามีหลายฟังก์ชั่นให้ใช้งาน ดังนั้นจึงต้องระวังที่จะไปใช้ขาพิเศษเหล่านั้น เช่น ESP8266 ที่ไม่สามารถนำ GPIO0 / GPIO15 / GPIO2 มาใช้งานได้เต็มรูปแบบ จากเดิมที่ขา GPIO มีเพียง 9 ขา (ไม่นับ TX0 RX0) เมื่อตัดขาทั้ง 3 ขาออกไป ทำให้เหลือขาใช้งานเพียง 6 ขาเท่านั้น ซึ่งไม่พอสำหรับใช้งาน Keypad และถึงต่อ Keypad ได้ ก็จะไม่มีสายเพียงพอไปต่อเซ็นเซอร์อื่น ๆ หรือจอแสดงผลได้

วิธีแก้ไข จึงเป็นการลดขาที่ไปต่อเข้ากับไมโครคอนโทรลเลอร์ลง มีทั้งวิธีนำตัวต้านทานมาต่อเพิ่มเพื่อให้เวลากดปุ่มแต่ละปุ่มค่าแรงดันเอาต์พุตที่อ่านได้จะแตกต่างกัน หรือใช้ไอซีขยายพอร์ตขนาดใหญ่ หรือใช้ Shift Register วิธีเหล่านี้มองว่าทำงานไม่เสถียร หรือ สิ้นเปลืองค่าใช้จ่ายมากเกินไป

ก่อนนำไปใช้งานจะต้องทราบหมายเลข Address ของโมดูลขยายพอร์ต PCF8574 เสียก่อน หลังจากต่อขาเพื่อทำการสื่อสารแบบ I2C กับไมโครคอนโทรลเลอร์แล้วสามารถใช้โค้ดที่มีชื่อว่า Arduino I2C Scanner เพื่อทำการค้นหาหมายเลข Address

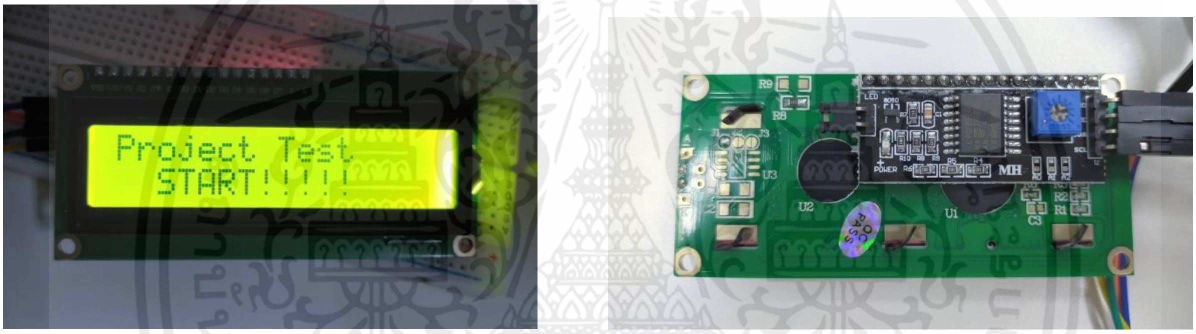
```
COM5

13:43:22.842 -> Scanning...
13:43:22.842 -> I2C device found at address 0x20
13:43:22.842 -> done
13:43:22.842 ->
13:43:27.845 -> Scanning...
13:43:27.845 -> I2C device found at address 0x20
13:43:27.845 -> done
13:43:27.845 ->
```

รูปที่ 2.20 หมายเลข Address ของโมดูลขยายพอร์ต PCF8574

2.2.12 Liquid Crystal Display with I2C Interface

สำหรับโครงงานนี้จะใช้จอแสดงผล LCD ชนิด Text Mode ขนาด 4 บรรทัด 20 ตัวอักษร ทำหน้าที่ปรากฏข้อความต้อนรับผู้มาเยือนและแสดงเบอร์โทรศัพท์ที่ได้รับจากการกด Keypad การใช้งาน LCD แบบปกติต้องใช้สายสัญญาณ หรือ Pin ของ Arduino เยอะมาก ทำให้เปลือง Pin ที่อาจจะต้องไปใช้งานกับอุปกรณ์ต่อพ่วงอื่นๆ ดังนั้นจะมีคนนำเอา I2C มาประยุกต์ใช้งานกับ LCD โมดูล เพื่อที่จะลดจำนวน Pin ที่ต้องใช้จำนวนมาก เหลือเพียง 2 สาย คือ SCL และ SDA เท่านั้น นอกจากนี้ยังคงสามารถติดต่อกับอุปกรณ์อื่นที่ใช้ I2C ได้บนสายเดียวกันด้วย

จอ LCD ที่มีการเชื่อมต่อแบบ I2C หรือเรียกอีกอย่างว่าการเชื่อมต่อแบบ Serial จะเป็นจอ LCD ธรรมดาที่มาพร้อมกับบอร์ด I2C Bus ที่ทำให้การใช้งานได้สะดวกยิ่งขึ้นและยังมาพร้อมกับ VR สำหรับปรับความเข้มของจอ ในรูปแบบ I2C จะใช้ขาในการเชื่อมต่อกับ Microcontroller เพียง 4 ขา (แบบ Parallel ใช้ 16 ขา) ซึ่งทำให้ใช้งานได้ง่ายและสะดวกมากยิ่งขึ้น

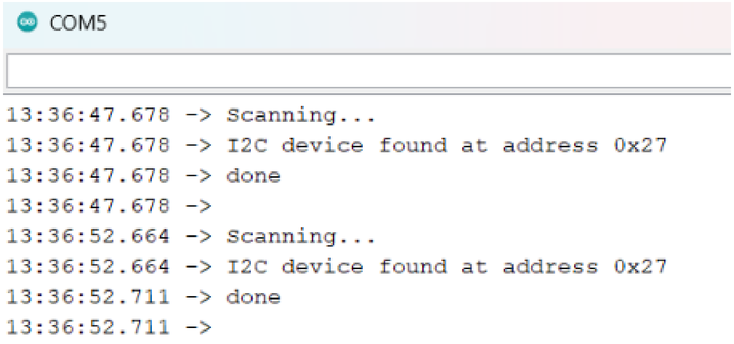


รูปที่ 2.21 I2C 20x4 LCD

ในการควบคุมหรือสั่งงาน สามารถส่งรหัสคำสั่งสำหรับควบคุมการทำงานของจอ LCD (I2C) เช่นเดียวกันกับจอ LCD แบบธรรมดา แต่ต่างกันตรงที่รูปแบบในการรับส่งข้อมูล การส่งข้อมูลรูปแบบ I2C ที่ใช้ขา 4 ขา ที่ใช้ในการเชื่อมต่อเท่านั้น

1. GND เป็น Ground ใช้ต่อระหว่าง Ground ของระบบ Microcontroller กับ LCD
2. VCC เป็นไฟเลี้ยงวงจรที่ป้อนให้กับ LCD มีขนาด +5VDC
3. SDA (Serial Data) เป็นขาที่ใช้ในการรับส่งข้อมูล
4. SCL (Serial Clock) เป็นขาสัญญาณนาฬิกาในการรับส่งข้อมูล

ก่อนนำไปใช้งานจะต้องทราบหมายเลข Address ของโมดูล I2C LCD เสียก่อน ทำเหมือนกับวิธีค้นหาหมายเลข Address ของโมดูลขยายพอร์ต PCF8574

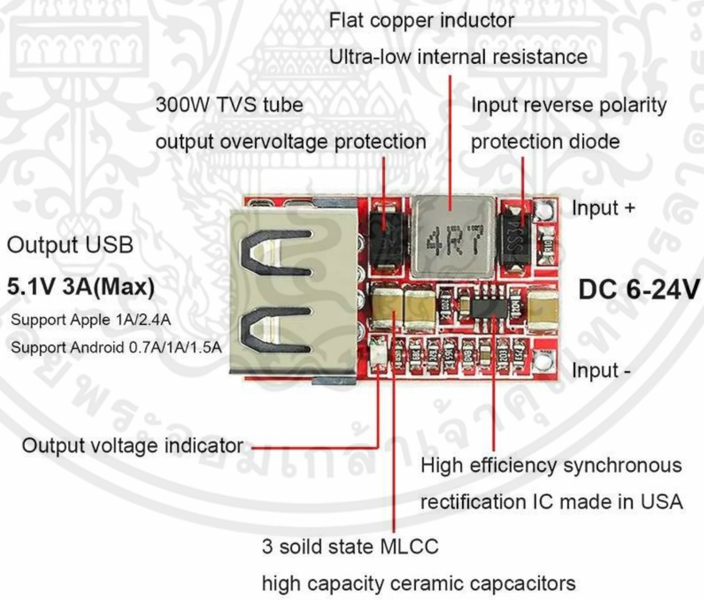


รูปที่ 2.22 หมายเลข Address ของโมดูล I2C LCD

2.2.13 DC to DC Buck (Step down) converter

วงจร Buck Converter หรือ วงจร Step down converter เป็นวงจรที่ลดแรงดันไฟฟ้าให้ต่ำลงเพื่อเหมาะสมต่อการใช้งาน โครงสร้างของวงจรจะประกอบด้วยสวิทช์ที่สามารถสั่งให้ "ON" หรือ "OFF" ได้ทุกขณะตามความต้องการในทางปฏิบัติสวิทช์ที่ใช้ในวงจรจริงคืออุปกรณ์สวิทช์ซึ่งเช่น BJT , MOSFET นอกจากนี้ยังมีส่วนประกอบอื่นๆอีกเช่น Condenser , Inductance และ Diode เป็นต้น

ในโครงโครงงานนี้จะใช้วงจร Buck converter ที่สามารถลดแรงดันไฟจากแบตเตอรี่ 12V 3A ให้เหลือ 5V เพื่อจ่ายไฟให้กับเซนเซอร์และไมโครคอนโทรลเลอร์



รูปที่ 2.23 6-24V DC to DC-5V Buck Converter

[ที่มา https://s.lazada.co.th/s.jdkGc]

โดยโมดูลดังกล่าวมีคุณสมบัติดังนี้

- Module Type: non-isolated step-down module (BUCK) synchronous rectification
- Input voltage: DC 6-24 V

- Output voltage: 5.1-5.2V
- Output current: up to 3 A
- Conversion efficiency : up to 97.5%
- Switching frequency : 500KHz
- Output ripple: 10mV (12V to 5V3A) 20M bandwidth
- Operating temperature: industrial grade (-40 ℃ to +85 ℃)
- Output overvoltage protection: Yes, input 1.5A fuse, output 5V 300W TVS tube clamp protection.
- Full load temperature rise: 30 ℃
- Quiescent current: 0.85 mA
- Load regulation: ± 1 %
- Voltage regulation: ± 0.5%
- Dynamic response speed: 5% 200uS

2.2.14 Sleep Mode

ด้วยคุณสมบัติของ ESP32 ที่เหมาะสำหรับทำอุปกรณ์ IoT ทั้งทำเป็น Node ต่าง ๆ และทำเป็น Gateway สำหรับการนำ ESP32 ไปทำเป็น Node มักจะมีปัญหาหนึ่งซึ่งเป็นปัญหาสำคัญของผู้พัฒนา คือ ปัญหาการใช้พลังงาน การสร้าง Node ส่วนใหญ่มักจะใช้แบตเตอรี่เป็นพลังงาน เพื่อให้ Node สามารถทำงานได้ แต่ด้วยคุณสมบัติของ ESP32 ที่เมื่อมีการใช้งาน WiFi จะใช้พลังงานไฟฟ้าจำนวนมาก เมื่อมีการใช้งานอย่างต่อเนื่องจะทำให้สิ้นเปลืองพลังงานมาก และทำให้แบตเตอรี่หมดอย่างรวดเร็ว ดังนั้น การสร้าง Node ของ ESP32 จึงควรคำนึงถึงการประหยัดพลังงานเป็นสำคัญ การประหยัดพลังงานที่ง่ายที่สุดคือการปิดใช้งานระบบต่าง ๆ ทั้ง WiFi และ CPU ทำให้ประหยัดพลังงานลงไปได้ สำหรับ ESP32 ได้เตรียมฟังก์ชั่นต่าง ๆ ไว้ให้ใช้งานแล้ว โดยถูกเรียกรวม ๆ ว่า Deep Sleep

โหมดการใช้งาน Deep Sleep จะแบ่งได้ตามวิธีการ “ปลุกให้ตื่น” โดยสามารถแบ่งลักษณะการปลุกให้ตื่นได้ 2 แบบ ดังนี้

ตื่นด้วยเหตุการณ์ภายนอก หมายถึง การทำให้ ESP32 กลับมาเริ่มทำงานใหม่ด้วยการกระตุ้นจากภายนอก แบ่งการกระตุ้นได้ 2 แบบ คือ

- ใช้เซ็นเซอร์สัมผัส – เหมาะสำหรับการใช้นิ้วสัมผัสโดยตรงเพื่อให้ ESP32 สามารถกลับมาทำงานได้
- เปลี่ยนสถานะ GPIO – ใช้เมื่อต้องการให้อุปกรณ์อื่นกระตุ้นทำให้เกิดการตื่นขึ้นมา เช่น การใช้งานเซ็นเซอร์ต่าง ๆ เพื่อกระตุ้นให้เกิดการส่งข้อมูลทันที

ตื่นด้วยเหตุการณ์ภายใน หมายถึง การใช้วงจรไฟฟ้าภายใน ESP32 เป็นตัวกระตุ้น แล้วทำให้ ESP32 ตื่นขึ้นมา มักจะหมายถึงการปลุกให้ตื่นตามเวลาด้วย RTC โดยปกติแม้ส่วนวงจรอื่น ๆ ของ ESP32 จะถูกปิด

การทำงาน แต่ส่วนของ RTC จะยังนับ และจดจำค่าเวลาต่อไปเรื่อย ๆ ตราบใดที่ยังมีการจ่ายพลังงานไฟฟ้าเลี้ยงอยู่ตลอด เพื่อรอการดึงค่าเวลาออกมาใช้งาน

ฟังก์ชั่นกำหนดการตื่น

2.2.15 ฐานข้อมูล

ฐานข้อมูล หรือ Database คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลายๆ แฟ้มข้อมูล ซึ่งถูกจัดเก็บอย่างเป็นระบบ โดยมีซอฟต์แวร์เข้ามาควบคุมกระบวนการใช้งาน การทำงาน หรือการประมวลผล ทำให้ผู้ใช้สามารถใช้ข้อมูลได้อย่างมีประสิทธิภาพ นอกจากนี้ยังมีหลายภาษาที่สามารถทำงานร่วมกันกับฐานข้อมูล MySQL ได้ อาทิ C,C++ , Python, Java และอื่นๆอีกมากมาย

โดยฐานข้อมูลที่มีประสิทธิภาพนั้นต้องมี

- Performance Database ที่มีประสิทธิภาพการทำงานสูง เพื่อรองรับงานได้หลากหลายรูปแบบภายในระบบเดียว
- Security มีความมั่นคงปลอดภัย ซึ่งสามารถปกป้องข้อมูลที่สำคัญได้ตลอดเวลา
- Availability มั่นคงและมีเสถียรภาพ มี Downtime ที่ต่ำ ซึ่งจะลดโอกาสที่ระบบจะหยุดทำงาน อีกทั้งต้องมีการอัปเกรดระบบทั้งในระดับของ Software และ Hardware ตลอดเวลา
- Manageability บริหารจัดการและบำรุงรักษาได้ง่าย เพื่อช่วยลดระยะเวลาในการทำงานลง
- Storage จัดเก็บข้อมูลได้อย่างเต็มประสิทธิภาพ สามารถลดหรือเพิ่มขยายได้ทั้งในระยะสั้นและระยะยาว

ประเภทของฐานข้อมูล แบ่งตามการจัดการ

SQL เป็นภาษาคอมพิวเตอร์ชนิดหนึ่ง ย่อมาจาก Structure Query Language มีการเชื่อมต่อข้อมูลที่มีความสัมพันธ์กัน (Rational) ซึ่งจะมีการสร้างตารางที่สามารถเชื่อมความสัมพันธ์ระหว่างข้อมูลได้ ตารางฐานข้อมูลประกอบด้วยคอลัมน์เป็นตัวบอกว่าเก็บข้อมูลประเภทใด และแถวเป็นรายละเอียดของข้อมูลของสิ่งๆหนึ่ง การนำไปประยุกต์ใช้ เช่น ระบบซื้อสินค้าออนไลน์ที่เรียกดูประวัติการซื้อได้ หรือระบบดูแลผู้ป่วยที่มีการบันทึกข้อมูลจำพวกส่วนสูง น้ำหนัก ความดันโลหิต และการเข้ารับการรักษา

No SQL ย่อมาจาก Not only SQL เป็นการเก็บข้อมูลที่ไม่จำเป็นต้องมีความสัมพันธ์กัน ลักษณะข้อมูลที่เข้ามาจะมีความยืดหยุ่นสูง ข้อมูลจะถูกจัดเก็บเป็นแนวนอนซึ่งสามารถกระจายข้อมูลไปยังเซิร์ฟเวอร์อื่นๆได้ การประยุกต์ใช้ เช่น ข้อมูลมหาศาลที่อาจจะไม่ได้เชื่อมโยงถึงกันแต่ต้องนำมาวิเคราะห์ เช่น การจราจรในช่วงเวลาต่างๆ หรือการทำระบบ IoT ที่ต้องรับค่าจากเซนเซอร์และแสดงขึ้นเว็บไซต์แบบเรียลไทม์ โดยหากต้องการเพิ่มลักษณะของข้อมูลในอนาคตแล้วการใช้ฐานข้อมูล No SQL ก็ย่อมตอบโจทย์เช่นกัน

ฐานข้อมูลที่ใช้ในโครงงานนี้คือ Firebase ซึ่งเป็นบริการหนึ่งจาก Google เป็นการบริการข้อมูลแบบออนไลน์ในรูปแบบ Real Time Database โดยการนำฐานข้อมูลที่มีมาแสดงใน Application หรือ Web

Application สาเหตุที่เลือกเครื่องมือนี้ในการทำโครงงานเนื่องจาก Firebase สามารถนำมาเป็นตัวกลางในการเชื่อมต่ออุปกรณ์เข้าด้วยกัน เช่น NodeMCU ส่งข้อมูลจากเซนเซอร์ไปยัง แอพพลิเคชั่นหรือเว็บไซต์ได้ มี API รองรับได้หลายภาษา เช่น Python สำหรับระบบปฏิบัติการบน Raspberry Pi , JavaScript สำหรับสร้างเว็บไซต์ , C/C++ สำหรับการพัฒนาบอร์ดไมโครคอนโทรลเลอร์บน Arduino IDE

รูปที่ 2.24 Firebase Realtime Database

[ที่มา https://elec2pcb.com/2021/11/11/firebase-la-gi-huong-dan-ket-noi-realtime-database-voi-node-red/]

2.2.16 เว็บแอปพลิเคชัน

Web Application หรือ เว็บแอป คือ คือแอปที่ถูกเขียนขึ้นมาให้สามารถเปิดใช้ใน Web browser ได้โดยตรง ไม่ต้องโหลด Application แบบเต็มๆ ลงเครื่อง ทำให้โดยรวมแล้วกินทรัพยากรค่อนข้างต่ำ สามารถเปิดใช้งานได้รวดเร็ว

ในปัจจุบันมี Web Application ที่หลากหลาย ทั้งในรูปแบบใช้งานระดับโลก ระดับมหาวิทยาลัย ไปจนถึงระดับบริษัท ก็มีเว็บแอปเป็นของตัวเองเช่นกัน เว็บแอพลิเคชัน ทำหน้าที่คล้ายกับเว็บไซต์ แต่จะสามารถเป็นแอปพลิเคชั่นให้ผู้คนใช้งานได้ด้วย ซึ่งจะต่างกับเว็บไซต์ที่เน้นให้ผู้คนเข้ามาดูเป็นหลัก เช่นเว็บแอปสำหรับคิดเลข เว็บแอปสำหรับจับเวลา เว็บแอปสำหรับแปลภาษา

ผู้ใช้งานจะใช้งาน Web Browser เพื่อทำการใช้งาน Web Application ซึ่งตัวเว็บแอปจะทำการดึงข้อมูลที่จำเป็นต้องใช้งานผ่าน Web Server และทาง Web server ก็อาจมีการดึงข้อมูลที่จำเป็น เช่นข้อมูลผู้ใช้ ข้อมูลลูกค้าผ่าน Database อีกทีหนึ่ง

โดยในโครงงานนี้ทำการพัฒนาเว็บแอพลิเคชันผ่านโปรแกรม VS Code โดยการเขียนโปรแกรมด้วยภาษาดังต่อไปนี้

HTML คือ ย่อมาจาก Hyper Text Markup Language คือภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบน website หรือที่เราเรียกกันว่าเว็บเพจ ถูกพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) และจากการพัฒนาทางด้าน Software ของ Microsoft ทำให้ภาษา HTML เป็นอีกภาษาหนึ่งที่ใช้เขียนโปรแกรมได้ หรือที่เรียกว่า HTML Application HTML เป็นภาษาประเภท Markup สำหรับการการสร้างเว็บเพจ โดยใช้ภาษา HTML สามารถทำโดยใช้โปรแกรม Text Editor ต่างๆ เช่น Notepad, Editplus หรือจะอาศัยโปรแกรมที่เป็นเครื่องมือช่วยสร้างเว็บเพจ เช่น Microsoft FrontPage, Dream Weaver ซึ่งอำนวยความสะดวกในการสร้างหน้า HTML ส่วนการเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะใช้โปรแกรม web browser เช่น IE Microsoft Internet Explorer (IE), Mozilla Firefox, Safari, Opera, และ Netscape Navigator เป็นต้น

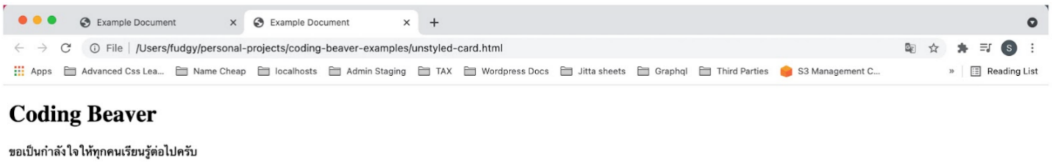
คำที่ศัพท์ที่เกี่ยวข้องกับ HTML

- Internet เครือข่ายคอมพิวเตอร์ที่ใหญ่ที่สุดในโลก เกิดจากการเชื่อมโยงของเครือข่ายต่างๆ เข้าด้วยกัน
- Hypertext รูปแบบเอกสารที่บรรจุการเชื่อมโยงไปยังเอกสารอื่นๆ ซึ่งสามารถใช้ข้อความ หรือรูป เป็นจุดเชื่อมโยง
- WWW ย่อจาก World Wide Web เป็นการสื่อสารด้วยการเชื่อมโยงเครือข่ายข่าวสารแบบใยแมงมุม (Web) แสดงผลด้วยเอกสารไฮเปอร์เท็กซ์
- HTTP ย่อมาจาก Hypertext Transfer Protocol เป็นรูปแบบการสื่อสารที่ใช้ในการรับส่งข้อมูลไฮเปอร์เท็กซ์ในเครือข่ายอินเทอร์เน็ต
- Web Browser โปรแกรมสำหรับแสดงผลหน้าเว็บ เช่น Internet Explorer, Mozilla Firefox และ Google Chrome เป็นต้น
- Web Page หน้าเอกสารที่อยู่ในรูปของไฮเปอร์เท็กซ์
- Web Site กลุ่มของหน้าเว็บหลายๆ หน้ารวมเข้าด้วยกัน
- Home Page หน้าเว็บ หน้าแรกของเว็บไซต์
- Web Site เครื่องให้บริการที่เป็นที่เก็บข้อมูลของเว็บไซต์

รูปแบบของภาษา HTML

```
<body>
  <div>
    <h1>Coding Beaver</h1>
    <p>ขอเป็นกำลังใจให้ทุกคนเรียนรู้ต่อไปครับ</p>
  </div>
</body>
```

ซึ่งจะได้การแสดงผลผ่านเว็บบราวเซอร์ดังนี้



รูปที่ 2.25 เว็บบราวเซอร์ที่ถูกเขียนด้วย HTML

[ที่มา https://blog.coding-beaver.com/first-thing-to-know-to-become-a-developer]

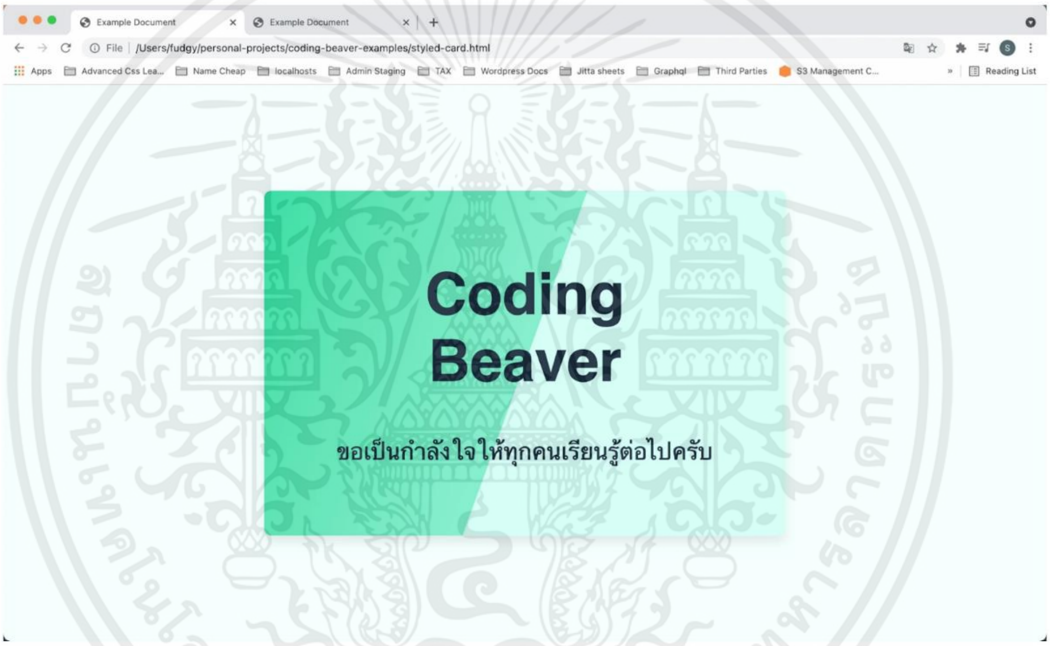
จะเห็นว่าหน้าเว็บไซต์ที่เขียนด้วยภาษา HTML อย่างเดียวยังไม่มีความสวยงามและสะดวกต่อผู้ใช้งาน ผู้พัฒนาสามารถตกแต่งหน้าเว็บไซต์เพิ่มได้ด้วยภาษา CSS

CSS ย่อมาจาก Cascading stylesheet เป็นอีกภาษาหนึ่งที่ Browser เข้าใจ ซึ่ง CSS มีหน้าที่ในการตกแต่ง HTML ให้สวยงาม

ตัวอย่างภาษา CSS

```
body {
  background-color: #f3fffd;
}
div {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  text-align: center;
  padding: 6rem;
  background: rgb(210,255,247);
  background: linear-gradient(290deg,
    rgba(210,255,247,1) 0%,
    rgba(210,255,247,1) 50%,
    rgba(126,237,202,1) 50%,
    rgba(39,218,156,1) 100%);
  border-radius: 10px;
  box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.1);
  color: #172A3A;
}
h1 {
```

```
  font-size: 5rem;
  font-weight: 600;
  font-family: sans-serif;
  margin-top: 0;
}
p {
  font-size: 2rem;
  font-family: sans-serif;
  margin-bottom: 0;
}
```



รูปที่ 2.26 เว็บบราวเซอร์ที่ถูกเขียนด้วย HTML และ CSS

[ที่มา https://blog.coding-beaver.com/first-thing-to-know-to-become-a-developer]

จะเห็นว่าหน้าเว็บไซต์ที่เขียนด้วยภาษา HTML และ CSS จะมีความสวยงามมากขึ้น

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช่ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียลเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)

สิ่งที่ JavaScript สามารถทำได้

- ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น
- มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่นเมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น นี่คือข้อดีของ JavaScript เลยก็ว่าได้ที่ทำให้เว็บไซต์ดังๆทั้งหลายเช่น Google Map ต่างหันมาใช้
- สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้แบบง่ายๆ นั่นเอง
- สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลืมกรอกอะไรบางอย่าง เป็นต้น
- สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ ใช้ web browser อะไร
- สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

หากจะสรุปโดยการเปรียบเทียบ ถ้าภาษา HTML เป็นเหมือนโครงสร้างของบ้าน ภาษา CSS ก็เปรียบเสมือนอุปกรณ์ในการตกแต่งบ้านหรือเฟอร์นิเจอร์ที่ทำให้บ้านสวยงามน่าอยู่ขึ้น และภาษา JavaScript เปรียบเสมือนระบบไฟฟ้า ระบบน้ำที่อยู่ภายในบ้านทำให้ผู้อาศัยมีสิ่งอุปโภคบริโภค

2.2.17 ไลน์โนติฟาย (Line Notify)

Line Notify คือบริการที่ LINE ให้เราส่งข้อความ หรือแจ้งเตือนอัตโนมัติ ไม่ว่าจะส่งเข้าผ่าน Group หรือบัญชีส่วนตัว ผ่าน API ของ LINE ได้โดยตรง

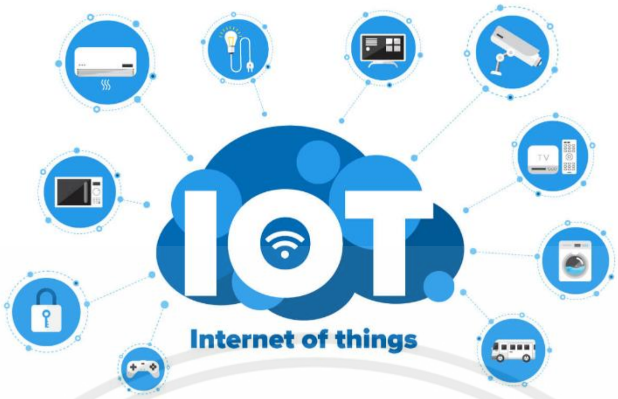
รูปที่ 2.27 Line Notify

[ที่มา https://scdn.line-apps.com/n/line_notice/img/og_160829/og_fb.png]

2.2.18 อินเตอร์เน็ตของสรรพสิ่ง (Internet of Things)

IoT (Internet of Things) หมายถึงเครือข่ายรวมของอุปกรณ์ที่เชื่อมต่อถึงกันและเทคโนโลยีที่อำนวยความสะดวกในการสื่อสารระหว่างอุปกรณ์กับระบบคลาวด์ ตลอดจนระหว่างอุปกรณ์ด้วยกันเอง จากการเกิดขึ้นของชิปคอมพิวเตอร์ราคาไม่แพงและการสื่อสารโทรคมนาคมที่มีแบนด์วิดท์สูง จึงทำให้ตอนนี้เรามีอุปกรณ์หลายพันล้านเครื่องที่เชื่อมต่อกับอินเทอร์เน็ต ซึ่งหมายความว่าอุปกรณ์ต่างๆ ในชีวิตประจำวัน เช่น

แปรงสีฟัน เครื่องดูดฝุ่น รถยนต์ และเครื่องจักรสามารถใช้เซ็นเซอร์เพื่อรวบรวมข้อมูลและตอบสนองต่อผู้ใช้ได้อย่างชาญฉลาด



รูปที่ 2.28 Internet of Things
[ที่มา https://www.globalsign.com/application/files/7416/3763/0034/General_Banner_WhatisIOT_4_APAC_2021_11_22.jpg, 2565]

องค์ประกอบของ Internet of Things
โดยทั่วไปจะมีองค์ประกอบดังนี้

1. Sensor สำหรับเป็นหน่วยรับข้อมูล ซึ่งอาจจะติดตั้งเพิ่มเติมในผลิตภัณฑ์ที่เคยมีอยู่แล้ว หรือเป็นผลิตภัณฑ์ใหม่ที่เพิ่งเกิดมาในโลกยุค Internet of Things หรือ Smart Device
2. การเชื่อมต่อเครือข่าย เพื่อให้ Sensor สามารถส่งข้อมูลไปยังระบบประมวลผลได้ อาจจะเป็นเครือข่ายภายใน หรือใช้เครือข่ายสาธารณะ
3. ระบบประมวลผลสำหรับรับข้อมูลจาก Sensor ชนิดเดียวกันหลายๆ ตัว หรือหลายๆ ชนิดก็ได้ เพื่อนำมาประมวลผล และส่งผลการวิเคราะห์ข้อมูลให้กับผู้ใช้งาน หรือส่งคำสั่งไปยังอุปกรณ์ใดๆ ก็ตาม ซึ่ง Big Data Analytics และ Cloud ก็เป็นอีกเทคโนโลยีที่เข้ามามีบทบาทในส่วนนี้อย่างมาก
4. ระบบบริหารจัดการสำหรับการเพิ่มอุปกรณ์เซนเซอร์ และระบบประมวลผลเข้ามาภายในระบบ Internet of Things และการติดตามการทำงาน การดูแลรักษา และการกำหนดค่าต่างๆ ของทุกๆ ส่วน ซึ่งบางครั้งระบบบริหารจัดการนี้ก็จะถูกรวมอยู่เข้ากับระบบประมวลผล
5. อุปกรณ์อื่นๆ เช่น อุปกรณ์ที่ไม่ได้ทำหน้าที่เป็นเซนเซอร์ แต่ทำการรับคำสั่งจากระบบ ประมวลผล เป็นต้น

2.2.19 โปรแกรม Arduino IDE

Arduino IDE คือ โปรแกรมที่ใช้ในการพัฒนางานสำหรับบอร์ด Arduino หรือ NodeMCU ในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด ซึ่งพัฒนาโดย Arduino.cc ขนาดของโปรแกรม Arduino IDE โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด

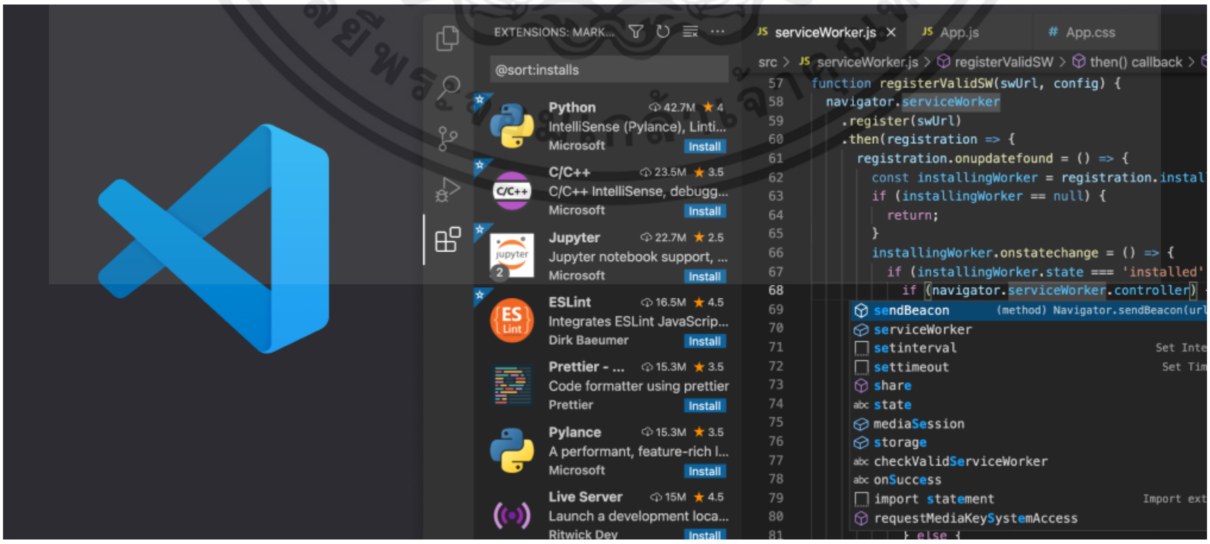
Arduino เข้าถึงฟังก์ชันเพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชั่นอื่น ๆ ของ Arduino IDE ย่อมาจาก Integrated Development Environment คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคน ที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็วถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น

รูปที่ 2.29 สัญลักษณ์โปรแกรม Arduino IDE

[ที่มา https://cdn.sparkfun.com/r/600-600/assets/learn_tutorials/6/1/arduinoThumb.jpg]

2.2.20 Visual Studio Code

Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ดจากค่ายไมโครซอฟท์ มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรี ๆ ที่ต้องการความเป็นมืออาชีพ ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาก ไม่ว่าจะเป็น 1.การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go 2.Themes 3.Debugger 4.Commands เป็นต้น



รูปที่ 2.30 โปรแกรม Visual Studio Code

[ที่มา https://code.visualstudio.com/]

บทที่ 3

วิธีการดำเนินงาน

3.1 อุปกรณ์ เครื่องมือ และโปรแกรมที่ใช้

1. ESP32-CAM
2. FTDI programmer
3. PIR motion sensor
4. Solenoid door lock
5. Relay
6. Matrix 4x4 Keypad
7. LCD with I2C interface

8.PCF8574 I/O expander module

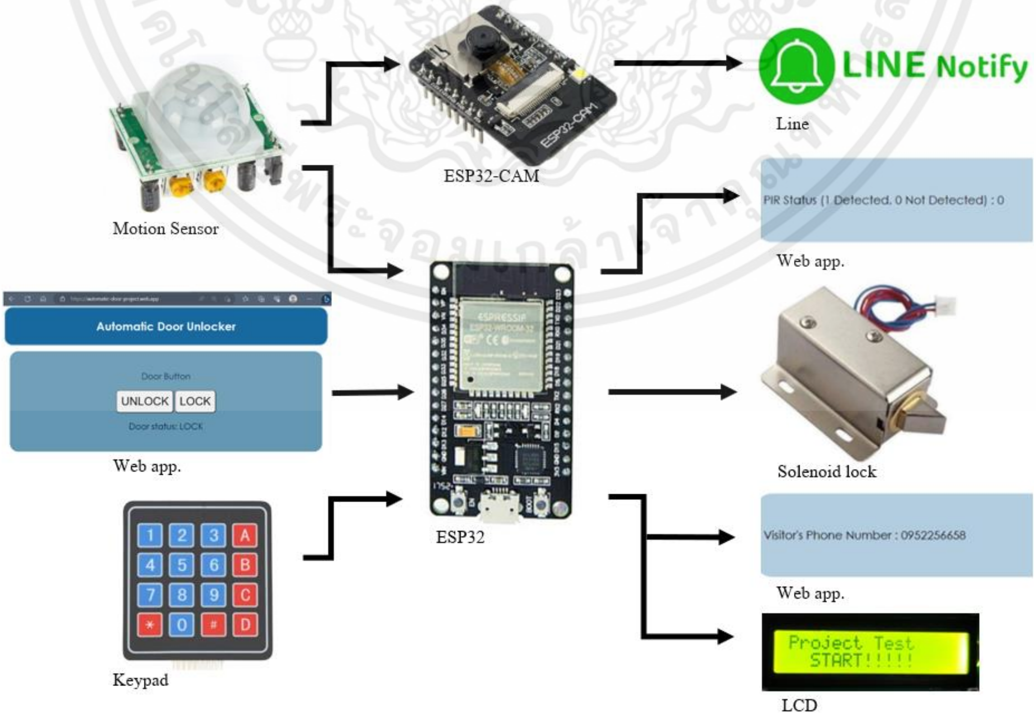
9.แบตเตอรี่ลิเธี่ยม

10.โปรแกรม Arduino IDE

11.ระบบเครือข่ายอินเตอร์เน็ต

12.โน้ตบุ๊ค

3.2 หลักการทำงานของระบบ



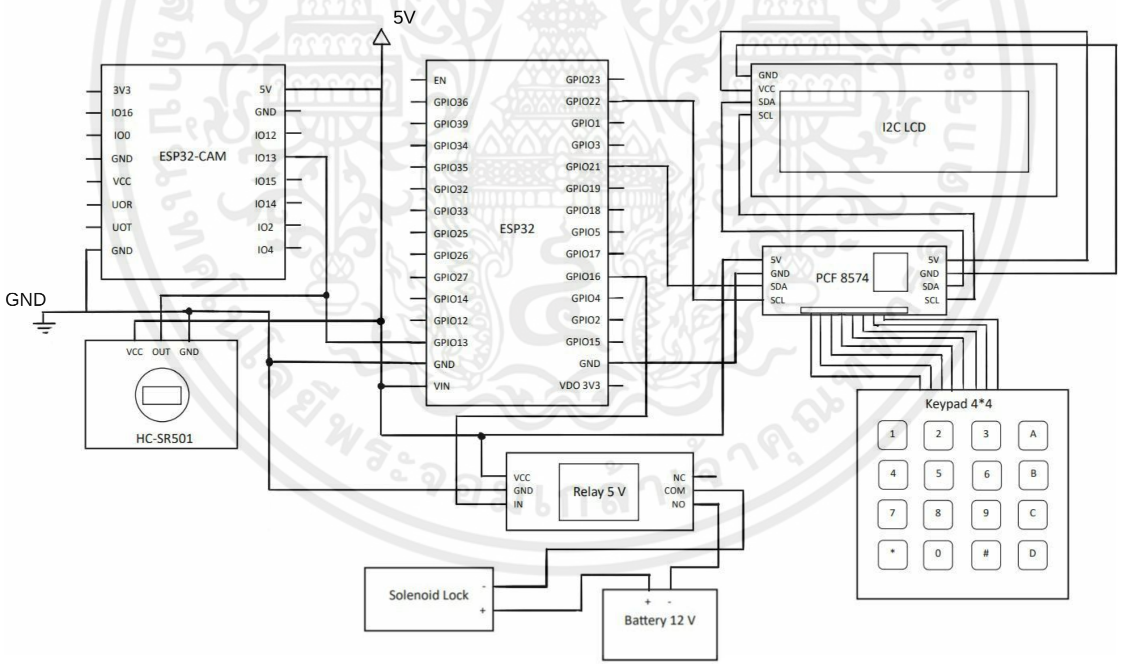
รูปที่ 3.1 Block diagram ของระบบ

Input คือ ส่วนที่รับสัญญาณไฟฟ้าจากอุปกรณ์อื่นเพื่อส่งให้ไมโครคอนโทรลเลอร์ทำการประมวลผล โดย Input ของระบบนี้ ประกอบด้วย Motion sensor, Web app. และ Keypad

Output คือ ส่วนที่สร้างสัญญาณส่งออก Output ของระบบตามคำสั่งที่เขียนให้กับไมโครคอนโทรลเลอร์ โดย Output ของระบบนี้ ประกอบด้วย Web app., Line notify, Solenoid lock และLCD

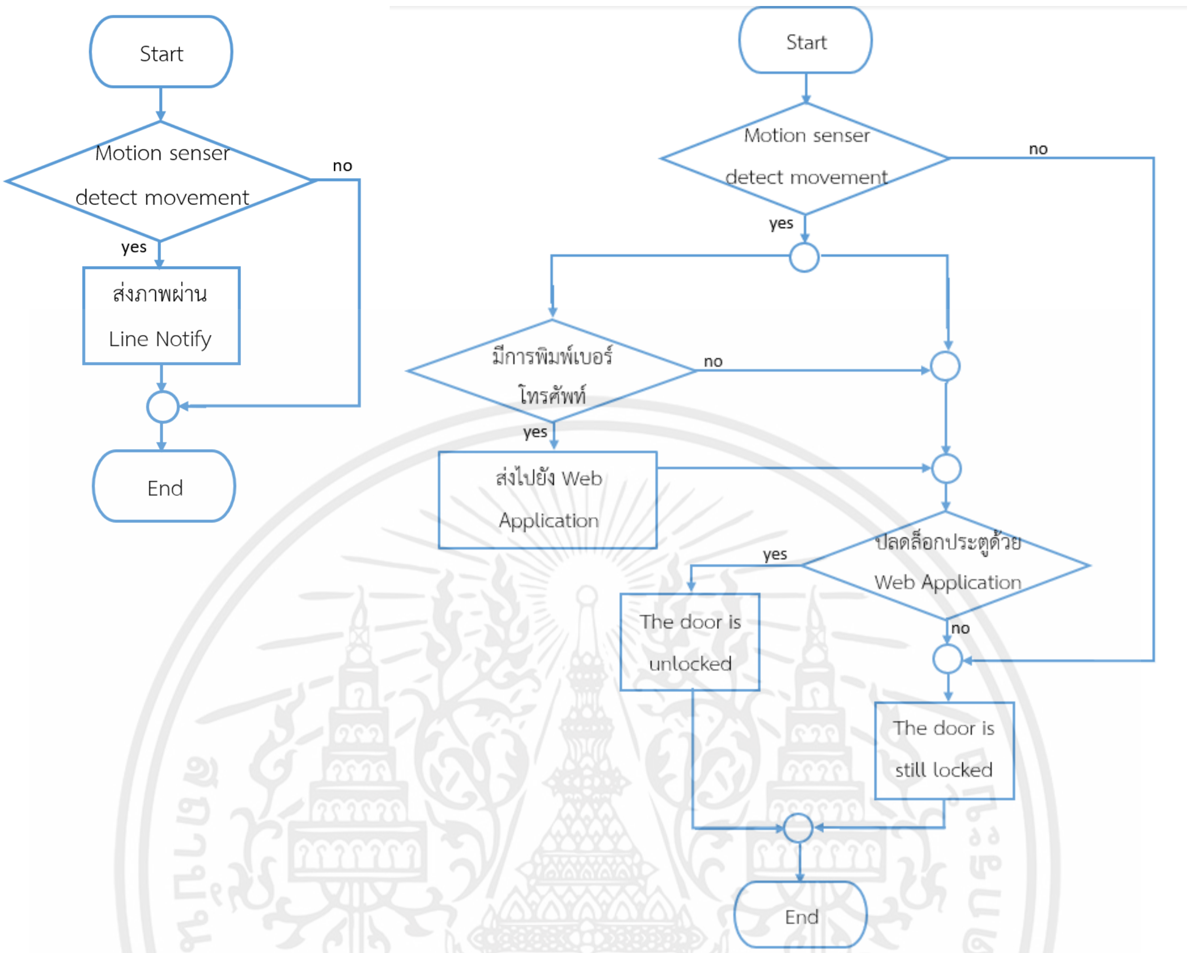
หลักการทำงานของระบบคือ Motion sensor จะทำการตรวจจับว่ามีสิ่งมีชีวิตเดินผ่านเซนเซอร์ในระยะและมุมที่กำหนดหรือไม่ ถ้ามีสิ่งมีชีวิตเดินผ่าน Motion Sensor จะส่งสัญญาณไฟฟ้าลอจิก 1 ให้กับไมโครคอนโทรลเลอร์ทั้งสองตัว เมื่อได้รับสัญญาณจาก Motion sensor ตัว ESP32-CAM จะสั่งการให้กล้องถ่ายภาพและส่งเข้าไปที่ Line Notify ของเจ้าของบ้าน เพื่อให้เจ้าของบ้านตรวจสอบได้ว่ามีใครมาบ้านหรือไม่ผ่านโทรศัพท์มือถือ และตัว ESP32 จะสั่งการให้แสดงสถานะของ Motion sensor บน Web app. ต่อมาเป็นส่วนของการควบคุมการเปิดปิดประตูจะสั่งการผ่านทาง Web app. สัญญาณจะถูกส่งมาที่ ESP32 และส่งต่อไปยัง Solenoid lock เพื่อให้เจ้าของบ้านตัดสินใจว่าจะเปิดประตูให้แก่ผู้มาเยือนหรือไม่ หากผู้มาเยือนต้องการติดต่อกับเจ้าของบ้าน สามารถพิมพ์เบอร์โทรศัพท์ผ่านทาง Keypad แล้วกดส่งมาได้ เบอร์โทรศัพท์ที่ถูกพิมพ์เข้ามาจะไปแสดงให้เจ้าของบ้านเห็นอยู่บน Web app.

3.3 Schematic



รูปที่ 3.2 Schematic

3.4 Flowchart

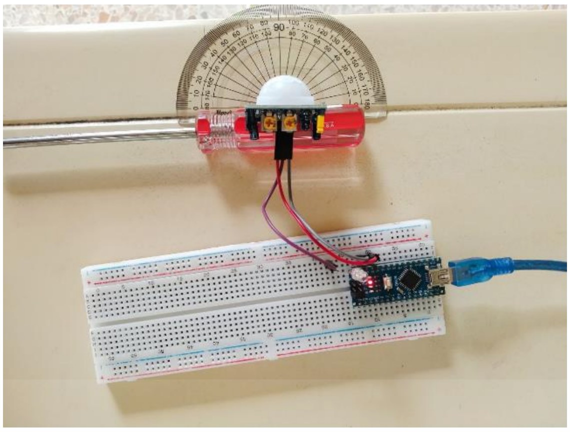


รูปที่ 3.3 Flowchart ของระบบ

3.5 การทดสอบ PIR Sensor กับมุมที่สามารถตรวจจับได้ที่ระยะทางต่างๆ

เนื่องจาก Data sheet ของโมดูล PIR Sensor รุ่น HC-SR501 กล่าวไว้ว่าสามารถตรวจจับวัตถุได้ในระยะมุมสูงสุด 120 องศา แต่ไม่ได้ระบุเป็นตัวเลขที่แน่นอนในแต่ละตำแหน่งระยะทางจาก PIR Sensor ผู้จัดทำโครงงานจึงอยากทราบว่าโมดูลที่นำมาใช้สามารถตรวจจับได้ มีขั้นตอนดังนี้

1. ต่อ PIR Sensor รุ่น HC-SR501 เข้ากับ LED สีน้ำเงิน และแหล่งจ่ายไฟ 5V (ใช้แหล่งจ่ายจาก Arduino Nano) แล้วใช้ไม้โปรแทรกเตอร์วางไว้เพื่อใช้ในการวัดมุมของเซนเซอร์

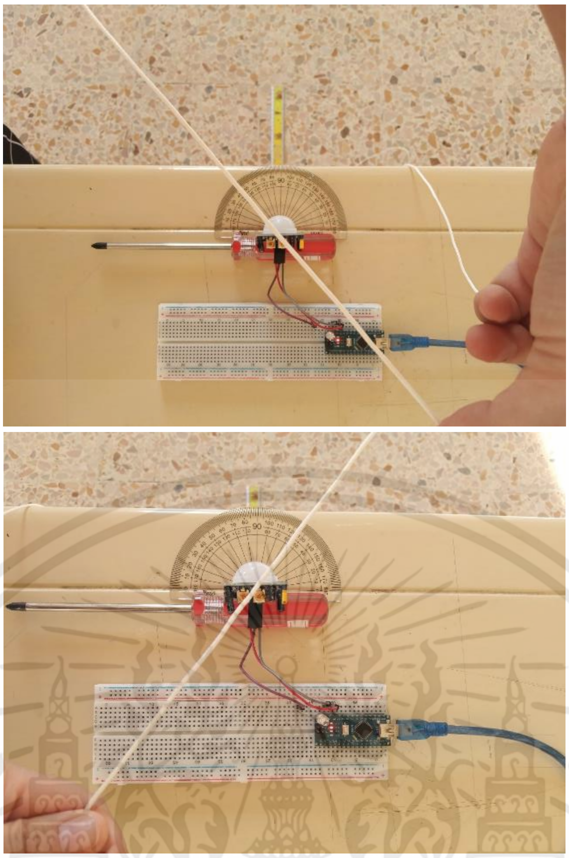


รูปที่ 3.4 การต่อ PIR Sensor รุ่น HC-SR501 เข้ากับ LED และแหล่งจ่ายไฟ

2. ปรับ Potentiometer ของ PIR Sensor ไปที่ระยะทางสูงสุด 3 เมตร จากนั้นให้คนลองเดินผ่านที่ระยะห่างจากเซนเซอร์เป็นระยะ 1 เมตร 2 เมตร และ 3 เมตร โดยเดินจากทางซ้ายไปขวา และจากทางขวาไปซ้าย

รูปที่ 3.5 ลักษณะการเดินผ่าน PIR Sensor

3. ทำการวัดมุมโดยการให้คนที่เดินถือเชือกเอาไว้เมื่อเซนเซอร์ตรวจจับการเคลื่อนไหวได้ LED จะติดไฟสีน้ำเงิน ผู้ทำการทดลองจะหยุดเดินแล้วอีกคนดึงเชือกผ่านจุดตรงกลางของเซนเซอร์



รูปที่ 3.6 วิธีการวัดมุมของเซนเซอร์

4. นำค่ามุมที่อ่านได้จากการเดินผ่านจากสองทางมาลบกัน ก็จะได้ขอบเขตมุมตรวจจับของเซนเซอร์ใน ระยะ 1 เมตร 2 เมตร และ 3 เมตร
5. ทำการวัดมุมซ้ำระยะละ 3 ครั้ง แล้วนำมาหาค่าเฉลี่ยของขอบเขตที่วัดได้

3.6 การทดสอบ Relay

3.6.1 ทดสอบหน้าสัมผัสของ Relay ว่าถูกต้องตามที่โมดูลระบุไว้หรือไม่ มีวิธีการทดสอบดังนี้

1. ใช้มัลติมิเตอร์วัดโวลต์ของรีเลย์ ระหว่างขา COM กับ NC โดยที่สองขานี้จะต้องเชื่อมถึงกัน
2. ใช้มัลติมิเตอร์วัดโวลต์ของรีเลย์ ระหว่างขา COM กับ NO โดยที่สองขานี้จะต้องไม่เชื่อมถึงกัน

3.6.2 ทดสอบว่า Relay มีลักษณะการทำงานแบบใด มีวิธีการทดสอบดังนี้

1. ต่อขา VCC และ GND ของรีเลย์เข้ากับแหล่งจ่ายไฟ 5VDC (ไฟ LED สีแดงจะติด)

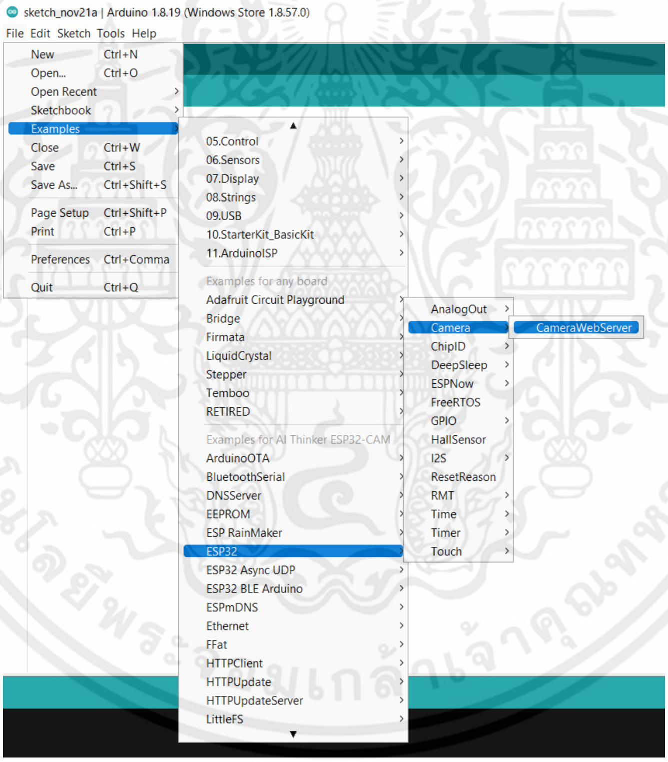
2. ต่อขา IN เข้ากับขาไฟเลี้ยง 5V หรือ GND

- หากต่อกับขาไฟเลี้ยง 5V แล้ว LED สีเขียวติด แสดงว่าเป็นการทำงานแบบ Active High
- หากต่อกับขา GND แล้ว LED สีเขียวติด แสดงว่าเป็นการทำงานแบบ Active Low

รูปที่ 3.7 โมดูลรีเลย์ที่นำมาทดสอบ

3.7 การทดสอบ ESP32-CAM ก่อนนำไปใช้งาน

โดยการต่อไมโครคอนโทรลเลอร์เข้ากับโมดูลอัพโหลด จากนั้นเข้าไปที่ Arduino IDE ไปที่ File > Examples > ESP32 > Camera > CameraWebServer



รูปที่ 3.8 การเรียกใช้โค้ดสำหรับดู Video Streaming

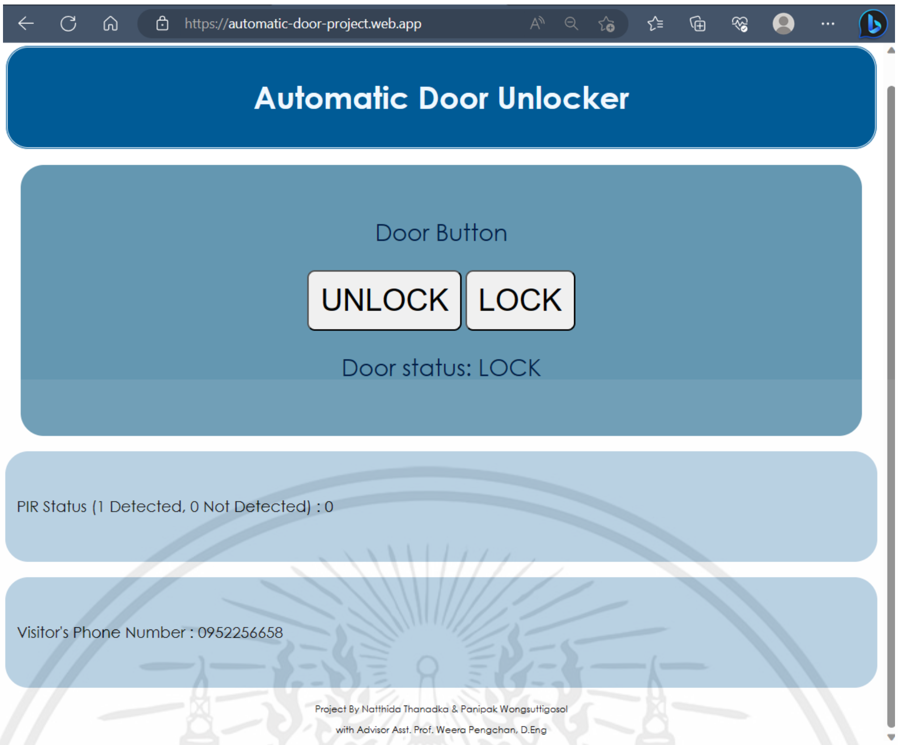
รูปที่ 3.9 ตัวอย่างโค้ด CameraWebServer

จากรูปที่ 3.9 เมื่อหน้าโค้ดปรากฏขึ้นมา ให้ Uncomment บรรทัดที่ 19 เพื่อเลือกรุ่นของกล้อง CAMERA_MODEL_AI_THINKER จากนั้นให้เปลี่ยนชื่อและรหัสของ Wifi ที่ต้องการเชื่อมตรงบรรทัดที่ 23 และ 24 แล้วทำการอัพโหลดโค้ด

3.8 การสร้างหน้าเว็บแอปพลิเคชันของโครงงาน

หลังจากที่ทำการ Deploy Website ที่สร้างขึ้นด้วย Firebase Web Hosting แล้ว ให้ไปที่ https://automatic-door-project.web.app/ จะปรากฏหน้าเว็บแอพซึ่งแบ่งออกเป็นกลุ่มบล็อค (Block) ได้แก่

- บล็อคสำหรับแสดงข้อความ ‘Automatic Door Unlocker’
- บล็อค ‘Door Button’ จะมีปุ่มสำหรับควบคุมการเปิดปิดของประตู และมีข้อความ Door Status สำหรับแสดงสถานะของประตู ณ ขณะนั้น
- บล็อคข้อความ PIR Status สำหรับแสดงสถานะของ PIR Sensor
- บล็อคข้อความ Visitor’s Phone Number สำหรับแสดงหมายเลขโทรศัพท์ของแขกผู้มาเยือน



รูปที่ 3.10 หน้าเว็บแอพลิเคชันของโครงงาน

บทที่ 4
ผลการทดลอง

4.1 ผลการทดสอบ PIR Sensor กับมุมที่สามารถตรวจจับได้ที่ระยะทางต่างๆ

จากการการทดสอบ PIR sensor กับมุมที่สามารถตรวจจับได้ในระยะต่างๆ มีผลดังแสดง ในตารางที่ 4.1

ตารางที่ 4.1 ผลทดสอบมุมของ PIR sensor

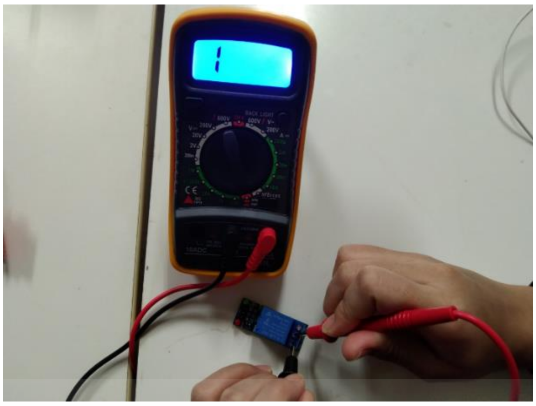
ระยะ(m) / การวัดครั้งที่	1	2	3
1	95 องศา	75 องศา	60 องศา
2	100 องศา	75 องศา	52 องศา
3	100 องศา	90 องศา	40 องศา
ค่าเฉลี่ย	98.3 องศา	80.0 องศา	50.6 องศา

4.2 ผลการทดสอบ Relay

เมื่อใช้มัลติมิเตอร์วัดระหว่างขา COM กับ NC พบว่ามัลติมิเตอร์ส่งเสียง แสดงว่าสายไฟเชื่อมถึงกัน

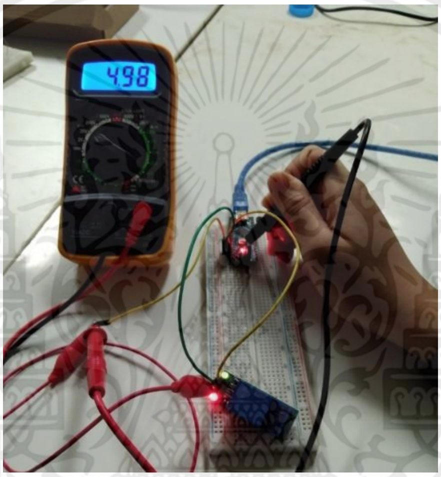
รูปที่ 4.1 การวัดค่าโวลต์ของรีเลย์ระหว่างขา COM กับ NC

เมื่อใช้มัลติมิเตอร์วัดระหว่างขา COM กับ NO พบว่ามัลติมิเตอร์ไม่ส่งเสียง แสดงว่าสายไฟไม่เชื่อมถึงกัน



รูปที่ 4.2 การวัดค่าโวลต์ของรีเลย์ระหว่างขา COM กับ NO

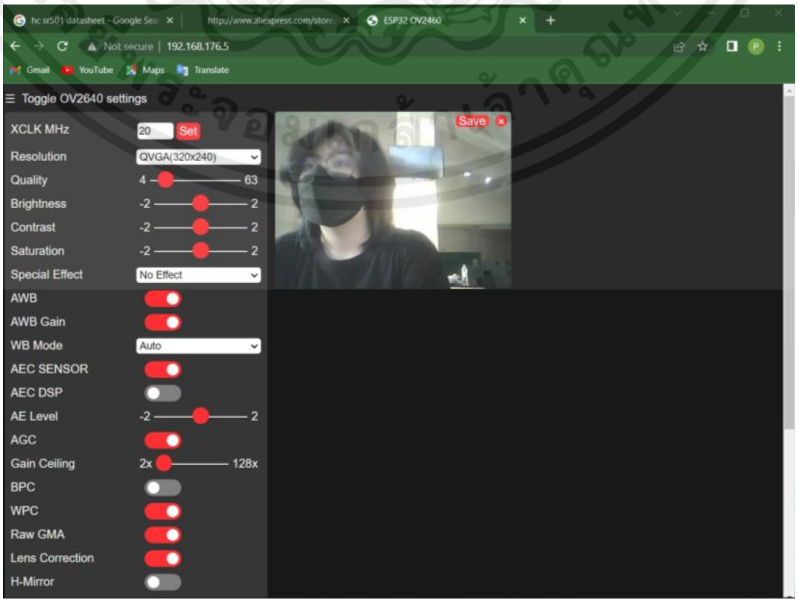
เมื่อต่อขา IN เข้ากับไฟเลี้ยง 5VDC พบว่า LED สีเขียวติด แสดงว่าเป็นการทำงานแบบ Active High



รูปที่ 4.3 การทดสอบประเภทการทำงานของรีเลย์

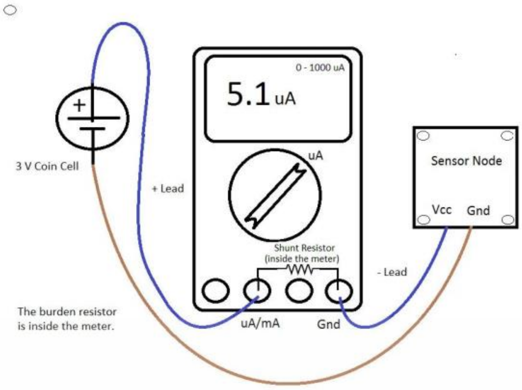
4.3 ผลการทดสอบ ESP32-CAM

เข้าไปที่ 192.168.176.5 เพื่อดูการทำงานของกล้องผ่านเซิร์ฟเวอร์ที่เป็นระบบ Video Streaming



รูปที่ 4.4 ดูการทำงานของโมดูลกล้องผ่าน Webserver

4.4 ผลการวัดกระแสไฟฟ้าของอุปกรณ์



รูปที่ 4.5 ไดอะแกรมที่ใช้ในการวัดกระแส

[ที่มา https://images.app.goo.gl/9o8fgo6RoWKVS3kP6]

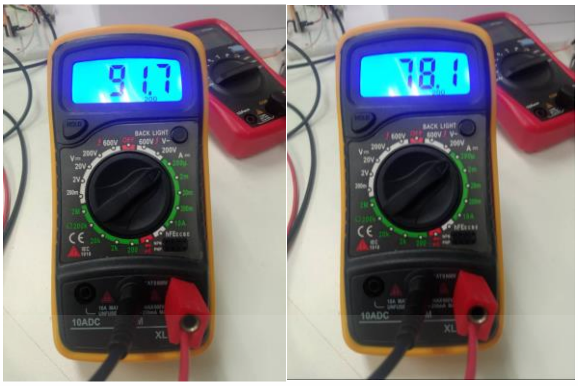
1. ESP32-CAM ตอน active mode กระเสไฟฟ้าเฉลี่ยอยู่ที่ 65.83 mA

รูปที่ 4.6 ปริมาณกระเสไฟฟ้าที่วัดได้เมื่อ ESP32-CAM active mode

2. ESP32-CAM ตอน deep sleep mode กระเสไฟฟ้าเฉลี่ยอยู่ที่ 4.57 mA

รูปที่ 4.7 ปริมาณกระเสไฟฟ้าที่วัดได้เมื่อ ESP32-CAM deep sleep mode

3. ESP32 ตอน active mode กระเสไฟฟ้าเฉลี่ยอยู่ที่ 84.9 uA



รูปที่ 4.8 ปริมาณกระเสไฟฟ้าที่วัดได้เมื่อ ESP32 active mode

4. ESP32 ตอน deep sleep mode กระเสไฟฟ้าเฉลี่ยอยู่ที่ 8.93 uA

รูปที่ 4.9 ปริมาณกระเสไฟฟ้าที่วัดได้เมื่อ ESP32 deep sleep mode

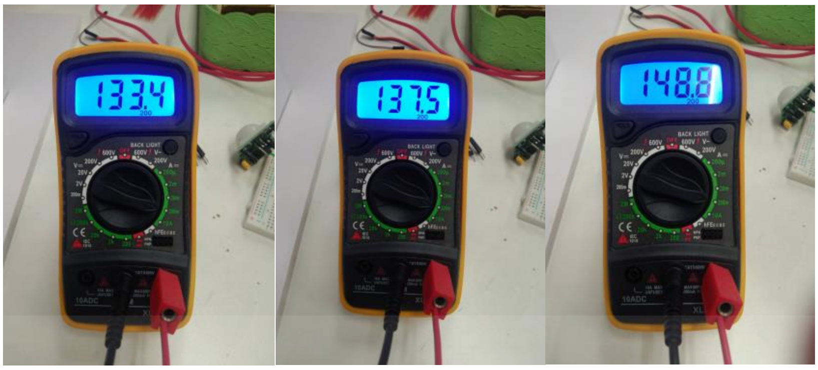
5. Keypad กระเสไฟฟ้าเฉลี่ยอยู่ที่ 30.83 uA

รูปที่ 4.10 ปริมาณกระเสไฟฟ้าที่วัดได้เมื่อกด Keypad

6. LCD กระเสไฟฟ้าเฉลี่ยอยู่ที่ 6.47 mA

รูปที่ 4.11 ปริมาณกระเสไฟฟ้าที่วัดได้เมื่อ LCD ทำงาน

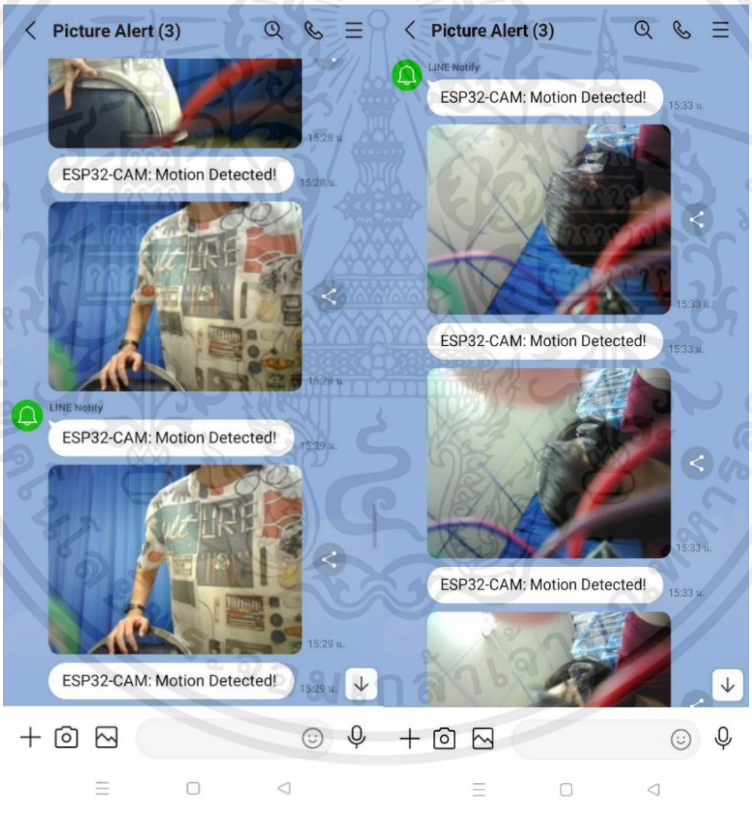
7. Solenoid lock กระแสไฟฟ้าเฉลี่ยอยู่ที่ 139.9 mA



รูปที่ 4.12 ปริมาณกระแสไฟฟ้าที่วัดจาก Solenoid lock

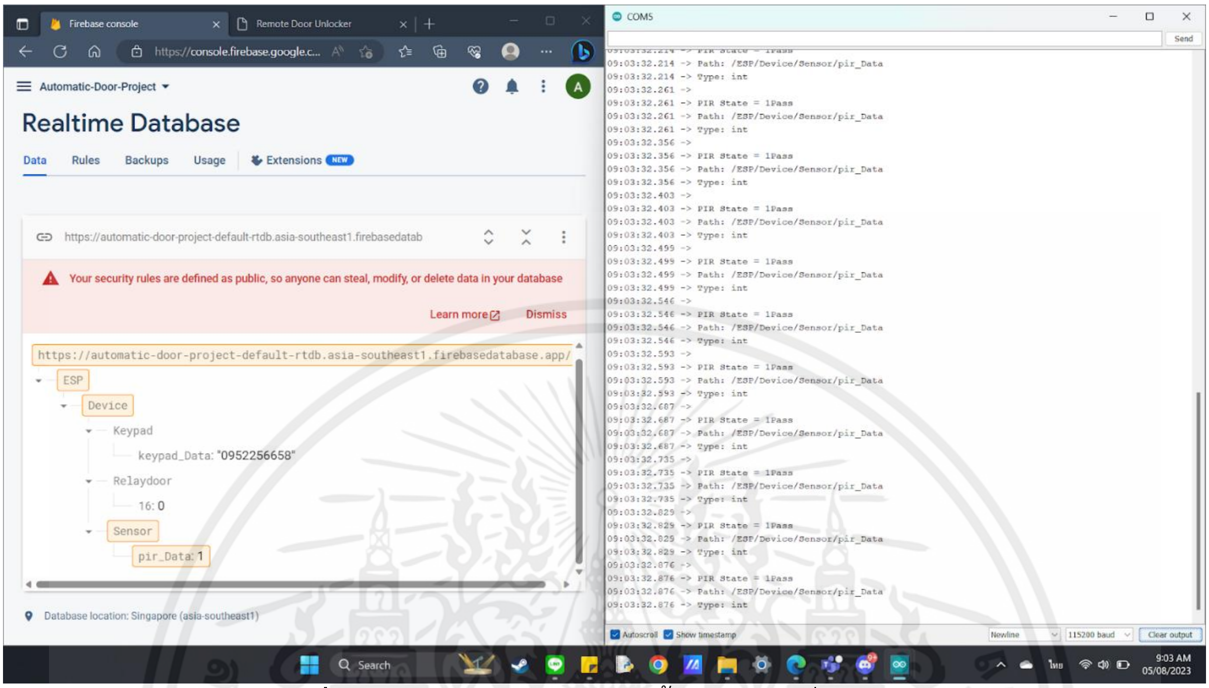
4.5 ผลของระบบปลดล็อคประตูอัตโนมัติ

- สามารถตรวจจับการเคลื่อนไหว และส่งแจ้งเตือนผ่านทาง LINE Notify มาที่โทรศัพท์มือถือของผู้ใช้ได้

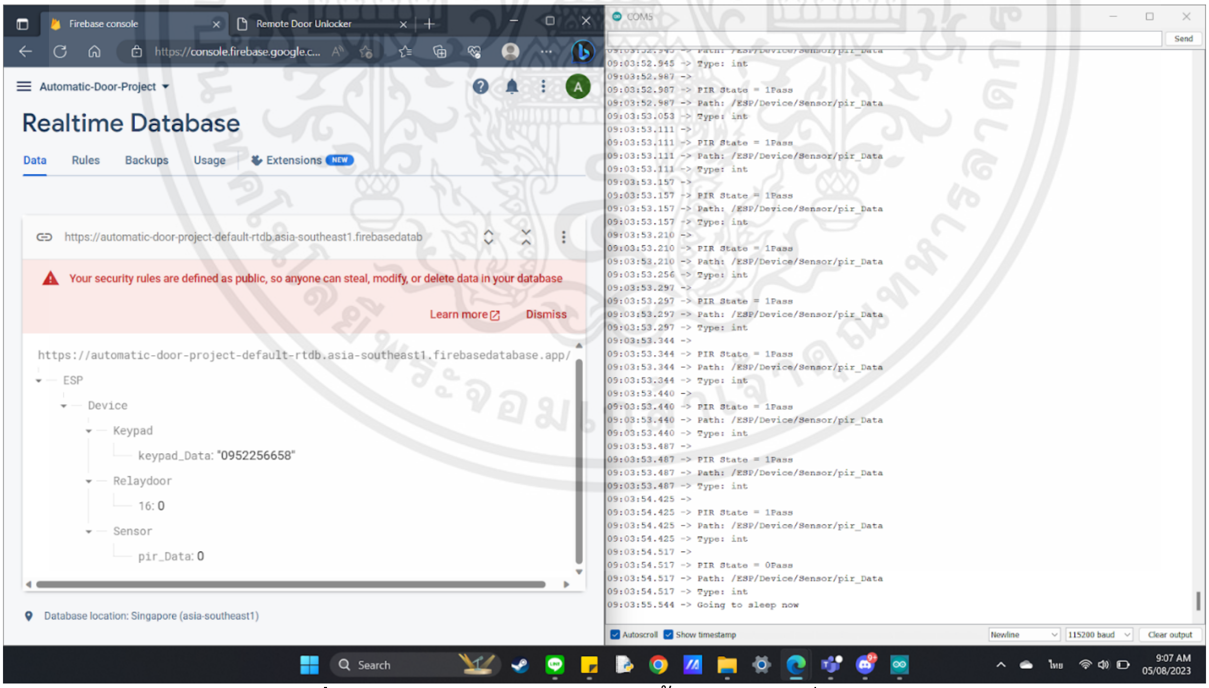


รูปที่ 4.13 การแจ้งเตือนผ่านทาง LINE Notify

- การส่งสถานะของ PIR sensor ผ่าน Path: /ESP/Device/Sensor/pir_Data ที่เชื่อมกับ Realtime Database เมื่อพบการเคลื่อนไหว PIR state = 1 ผ่าน Serial Monitor จากนั้นข้อมูลจะถูกส่งไปที่ Realtime Database ตาม Path ที่สร้างขึ้น เมื่อไม่พบการเคลื่อนไหว PIR state = 0 ผ่าน Serial Monitor



รูปที่ 4.14 Realtime Database ขึ้นสถานะ 1 ที่ pir_Data



รูปที่ 4.15 Realtime Database ขึ้นสถานะ 0 ที่ pir_Data

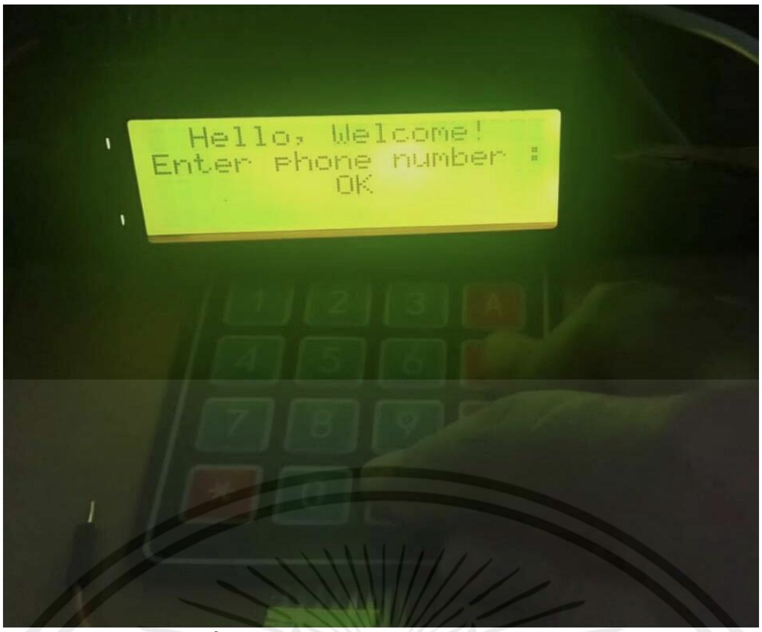
โดยในช่วงที่ PIR State = 0 แล้ว LCD ปรากฏข้อความ System Off ก่อนจะดับลงไป

รูปที่ 4.16 LCD ปรากฏข้อความ System Off

- การรับหมายเลขโทรศัพท์จากผู้มาเยือนปรากฏบน LCD และส่งไปที่ Realtime Database เมื่อผู้มาเยือนกดปุ่ม ‘C’ ซึ่งผู้จัดทำสื่อถึงข้อความ Contact LCD จะปราฏข้อความ ‘Hello, Welcome!’ ในแถวที่ 0 ปรากฏข้อความ ‘Enter phone number : ’ ในแถวที่ 1 จากนั้นหมายเลขโทรศัพท์ของผู้มาเยือนจะถูกแสดงผ่าน LCD ในแถวที่ 2

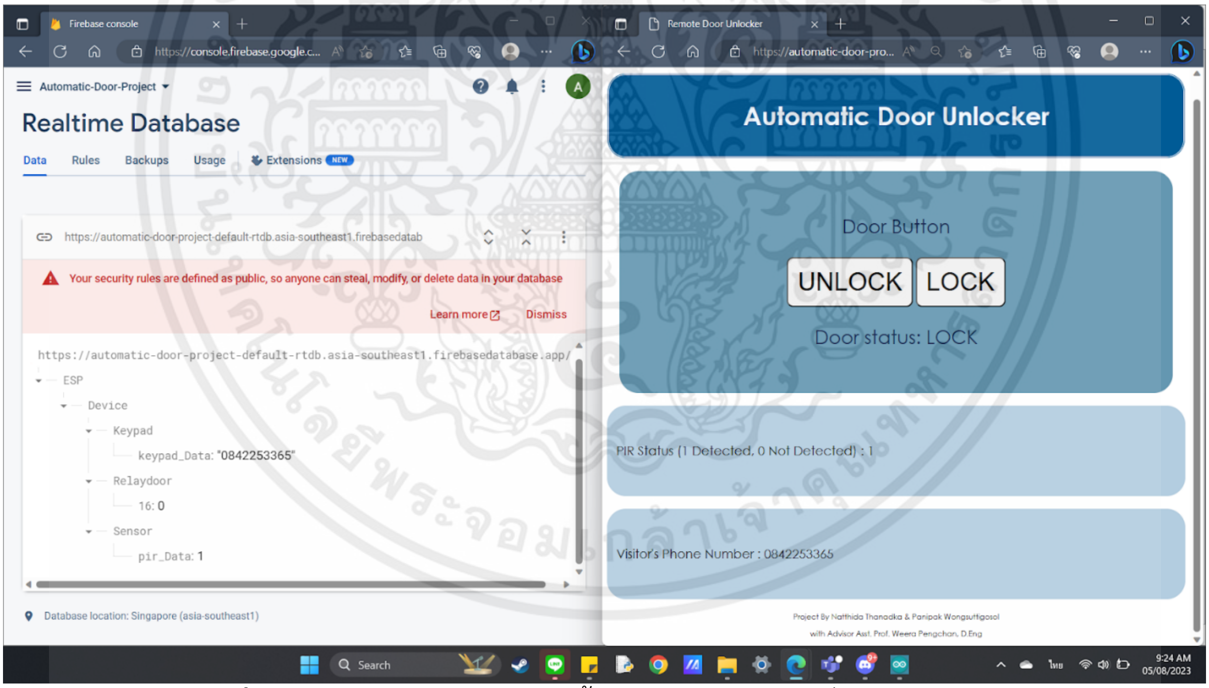
รูปที่ 4.17 LCD ปรากฏข้อความต้อนรับและหมายเลขโทรศัพท์

หากผู้มาเยือนตรวจสอบหมายเลขโทรศัพท์แน่ใจว่าถูกต้องแล้ว ให้ทำการกดปุ่ม ‘#’ จากนั้น LCD จะปรากฏข้อความ ‘OK’ ในแถวที่ 2 กรณีที่หมายเลขที่กรอกเข้าผ่าน Keypad มีจำนวน 10 หลัก แล้วส่งข้อมูลขึ้น Realtime Database



รูปที่ 4.18 LCD ปรากฏข้อความ ‘OK’

หมายเลขโทรศัพท์ถูกส่งไปที่เว็บแอพ ผ่าน Path: /ESP/Device/Keypad/keypad_Data



รูปที่ 4.19 Realtime Database ขึ้นหมายเลขโทรศัพท์ ที่ keypad_Data

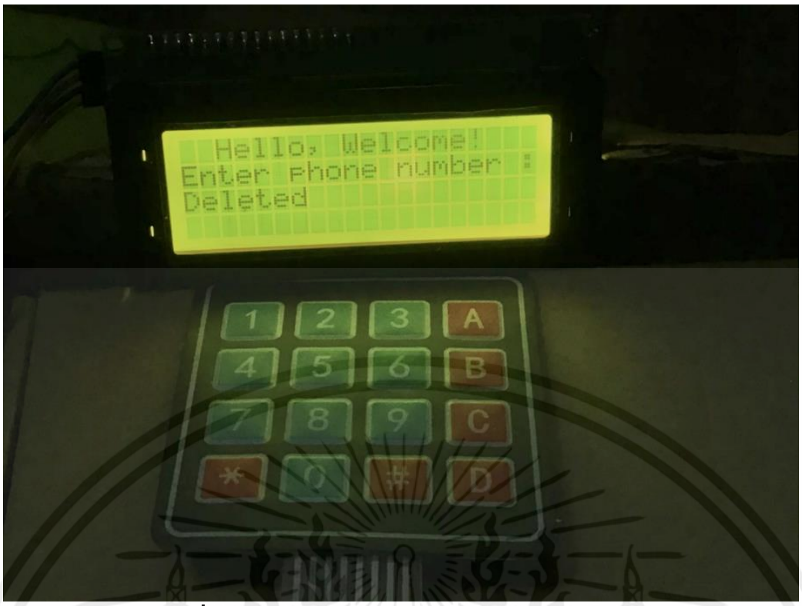
หรือสามารถดูเบอร์โทรศัพท์มือถือผ่านเว็บแอพบนโทรศัพท์มือถือได้เช่นกันเพียงเข้าไปที่บราวเซอร์แล้วไปที่ URL https://automatic-door-project.web.app/

รูปที่ 4.20 เจ้าของบ้านดูข้อมูลผ่านเว็บแอพในโทรศัพท์มือถือ

หากหมายเลขที่กรอกผ่าน Keypad มีจำนวนไม่เท่ากับ 10 หลัก LCD จะปรากฏข้อความ ‘Not Correct’ ในแถวที่ 2

รูปที่ 4.21 LCD ปรากฏข้อความ ‘Not Correct’

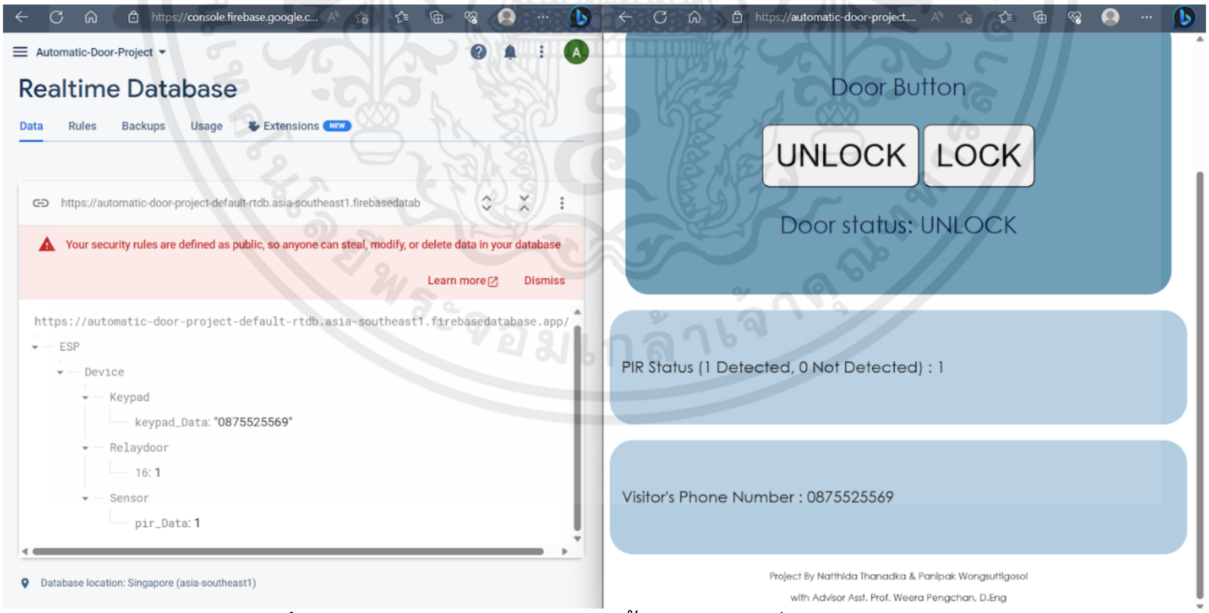
หากผู้มาเยือนต้องการกรอกหมายเลขโทรศัพท์อีกครั้งในกรณีที่กดปุ่มผิด ให้กดปุ่ม '*' LCD จะปรากฏข้อความ 'Deleted' ในแถวที่ 2 จากนั้นสามารถกรอกหมายเลขโทรศัพท์ใหม่ได้



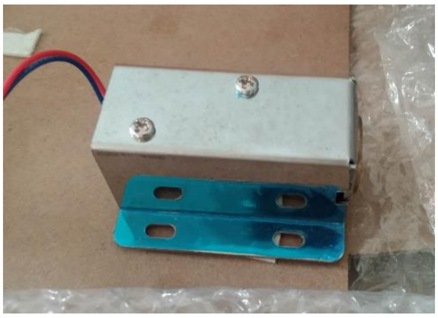
รูปที่ 4.22 LCD ปรากฏข้อความ 'Deleted'

- การควบคุมกลอนประตูไฟฟ้า Solenoid Door Lock ผ่าน Path: /ESP/Device/Relaydoor/16 ที่เชื่อมกับ Realtime Database

เมื่อกดปุ่ม UNLOCK ผ่านเว็บแอพ Relaydoor จะมีสถานะ 1 แล้วกลอนประตูจะถูกเปิด

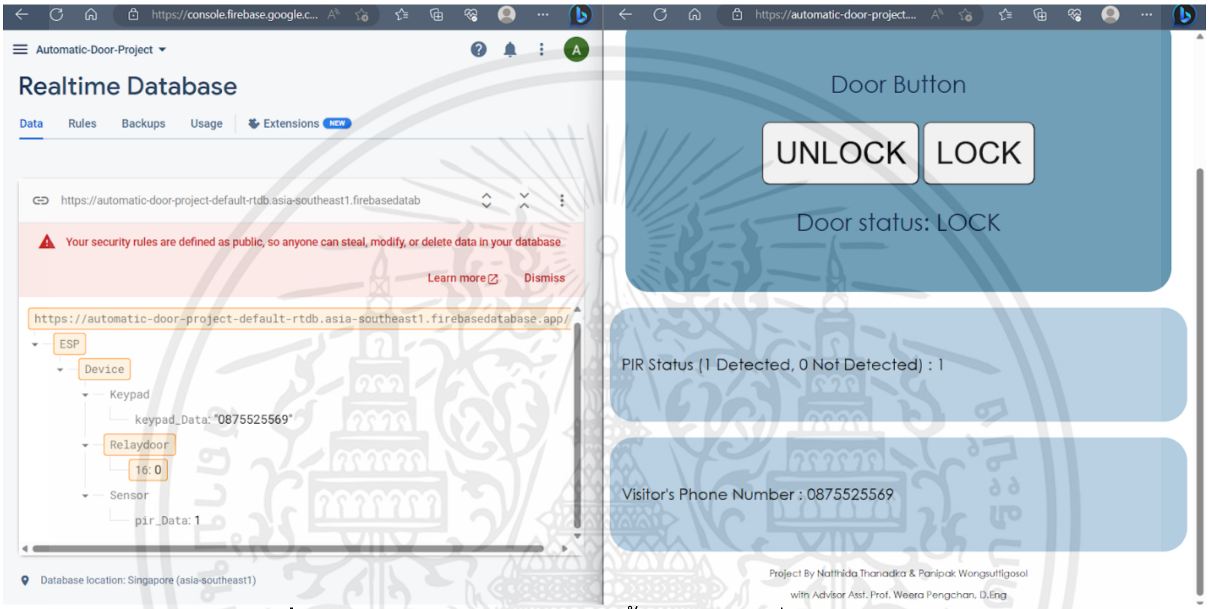


รูปที่ 4.23 Realtime Database ขึ้นสถานะ 1 ที่ Relaydoor/16



รูปที่ 4.24 กลอนประตูถูกเปิดตามการควบคุมจากเว็บแอพ

เมื่อกดปุ่ม LOCK ผ่านเว็บแอพ Relaydoor จะมีสถานะ 0 แล้วกลอนประตูจะถูกปิด



รูปที่ 4.25 Realtime Database ขึ้นสถานะ 0 ที่ Relaydoor/16

รูปที่ 4.26 กลอนประตูถูกเปิดตามการควบคุมจากเว็บแอพ

บทที่ 5
สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปและวิจารณ์ผลการทดลอง

จากการทดสอบแยกอุปกรณ์แต่ละตัว สามารถสรุปได้ดังนี้

การทดสอบ PIR Sensor ที่ระยะทางและมุมต่างๆ โดยปรับตัวต้านทานปรับค่าได้ไปที่ค่าต่ำสุด สามารถวัดระยะทางได้ถึง 3 เมตรได้ตามที่ Datasheet กำหนดไว้ นอกจากนี้ก็ได้ทำการทดสอบ Field of View ของเซนเซอร์ที่ใส่ Freshnel Lens ลงไป พบแนวโน้มว่าที่ระยะทางใกล้มุมตรวจจับจะกว้าง ส่วนที่ระยะไกลออกไปมุมตรวจจับจะแคบลง

การทดสอบ Relay พบว่าสามารถใช้งานได้ปกติ ซึ่งเป็นรีเลย์ประเภท Active High กล่าวคือจะต้องมีการส่งลอจิก 1 จากไมโครคอนโทรลเลอร์ออกไปเพื่อทำให้รีเลย์ทำงาน เวลาใช้งานในวงจรต้องต่อกับ NO เพื่อให้ในสภาวะปกติเป็นการเปิดวงจร แล้วใช้ไฟ 5V กระตุ้นหน้าสัมผัสย้ายไปที่ NC เพื่อปิดวงจรที่เชื่อมกับแหล่งจ่ายไฟด้าน 12V

การทดสอบโมดูลกล้อง ที่ติดกับ ESP32-CAM พบว่าสามารถใช้งานได้ปกติ สามารถถ่ายรูปหรือดู Streaming ได้อย่างที่ควรจะเป็น

ส่วนในการทดสอบระบบปลดล็อคประตูอัตโนมัติ โดยใช้ PIR Sensors เข้ามาช่วยตรวจสอบการเคลื่อนไหวของสิ่งมีชีวิต อีกนัยหนึ่งคือช่วยสังเกตว่ามีบุคคลใดมาเยี่ยมบ้านหรือไม่ จากนั้น ESP32-CAM จะถ่ายรูปแล้วส่งเข้า Line Notify จากนั้นการจะเปิดปิดประตูเพื่อให้คนเข้ามาในบ้านทำโดยการสั่งการผ่านเว็บแอพ โดยการเปิดปิดปุ่มกดแบบ Push-Pull เมื่อกลอนประตูล็อค (ไม่มีการจ่ายไฟ) สถานะของประตูจะแสดงข้อความ ‘LOCK’ และเมื่อกลอนประตูเปิด (มีการจ่ายไฟ) สถานะของประตูจะแสดงข้อความ UNLOCK ซึ่งสามารถทำให้ระบบทำงานตามที่ต้องการได้

สามารถปลดล็อคประตูผ่านเว็บแอพได้ตามที่ต้องการแม้เว็บแอพจะถูกใช้งานผ่านโทรศัพท์มือถือเมื่อใช้เครือข่ายอินเตอร์เน็ตต่างกัน นั่นหมายความว่าสามารถปลดล็อคประตูจากที่ไหนเมื่อใดก็ได้ตราบเท่าที่มีสัญญาณอินเตอร์เน็ต

ส่วนหน้าที่เสริมของโครงงานนี้ ได้แก่ การถ่ายรูปอัตโนมัติส่งเข้า Line nofity เมื่อ PIR sensor ตรวจพบการเคลื่อนไหว สามารถจับภาพได้ทันทีเมื่อพบการเคลื่อนไหว แต่พบว่ามีความล่าช้าประมาณ 10 วินาทีในการส่งรูปภาพไปยัง Line notify เจ้าของบ้าน

จอ LCD สามารถทำงานร่วมกับ Keypad ซึ่งเชื่อมต่อกับโมดูลขยายพอร์ต แสดงข้อความต้อนรับผู้มาเยือนเมื่อกดปุ่ม ‘C’ (Contact) ผู้มาเยือนสามารถพิมพ์เบอร์โทรศัพท์แล้วส่งไปให้เจ้าของบ้านเพื่อทำการ

สื่อสารกัน หลังจากที่กดปุ่ม ‘#’ (Enter) จะมีการตรวจสอบเงื่อนไขว่าเบอร์โทรศัพท์นั้นมีจำนวน 10 หลักหรือไม่ ถ้าใช่เบอร์โทรศัพท์นั้นจะถูกส่งไปที่ Realtime Database และส่งข้อมูลต่อไปที่เว็บแอพ ถ้าไม่เป็นไปตามเงื่อนไขจอ LCD จะขึ้นข้อความ ‘Not Correct, Try Again’ จากนั้นผู้มาเยือนสามารถกรอกเบอร์โทรศัพท์ใหม่ได้ นอกจากนี้ผู้มาเยือนสามารถกดลบที่ปุ่ม ‘*’ (Deleted) ในกรณีที่กรอกหมายเลขเบอร์โทรศัพท์ผิด จากนั้นข้อมูลในบรรทัดนั้นจะถูกล้างออกทั้งหมดผู้มาเยือนสามารถกรอกหมายเลขเบอร์โทรศัพท์ใหม่ได้อีกครั้ง

Keypad ไม่สามารถใช้ร่วมกับ ESP32-CAM ได้ แม้จะใช้โมดูลขยายขาสัญญาณ I2C หรือกำหนดขา I2C ผ่านฟังก์ชัน Wire.begin แล้วก็ตาม จึงให้ ESP32-CAM มีหน้าที่ถ่ายภาพอัตโนมัติเมื่อพบการเคลื่อนไหว ส่วนการแสดงผล LCD การรับเบอร์โทรจาก Keypad การควบคุมกลอนประตูจะถูกโปรแกรมผ่าน ESP32 Wroom

โดยในการทดลองได้พบปัญหาที่สำคัญ เช่น การอัพโหลดโค้ดเข้าไมโครคอนโทรลเลอร์ค่อนข้างมีความล่าช้า โดยในตอนที่ต่อลงบนบอร์ดทดลองก็ควรจะเช็คสายไฟให้ดีว่าเชื่อมติดหรือยัง หรืออาจจะเป็นที่การติดตั้งไลบรารี่ไม่ครบ ดังนั้นควรจะเผื่อเวลาในการอัพโหลดโปรแกรมและทำการปรับปรุงแก้ไขตัวโค้ด

นอกจากนี้การทดสอบและปรับค่าบนโมดูลและเซนเซอร์ต่างๆที่นำมาใช้ก็มีความสำคัญ ควรทำการศึกษา ทดสอบเชิงเทคนิคก่อนนำมาใช้แล้วจึงทำการเขียนโปรแกรมสั่งให้ระบบทำงาน

เรื่องสัญญาณอินเตอร์เน็ตก็มีความสำคัญที่จะทำให้ระบบทำงานได้ตามต้องการ หากสัญญาณไม่เสถียรก็จะทำให้ระบบทุกอย่างทำงานช้าลง

5.2 ข้อเสนอแนะ

1. ควรทำระบบยืนยันตัวตนด้วยการลงชื่อเข้าใช้ (Log in) เช่น กำหนดเงื่อนไขใน Firebase ให้เข้าถึงได้เฉพาะบุคคลที่กำหนดไว้เพื่อความปลอดภัยของระบบ

2. เนื่องจากว่าเซนเซอร์ที่ใช้สามารถตรวจจับสิ่งมีชีวิตทุกชนิดที่สามารถแผ่รังสีอินฟราเรด อาจจะมีระบบให้โมดูลกล้องถ่ายรูปเฉพาะตอนที่ตรวจจับใบหน้ามนุษย์ได้

3. อาจมีการบันทึกรูปภาพ ระบุวันเวลาที่มีคนมาบ้านลงใน Database เพื่อเก็บเป็นข้อมูลดูย้อนหลังได้ เนื่องจากการส่งรูปผ่าน Line Notify อาจจะมีวันหมดอายุและทำให้ไม่สามารถดูรูปย้อนหลังได้

4. ควรศึกษาเซ็นเซอร์ตัวอื่นๆเพิ่มเติม เพื่อเพิ่มประสิทธิภาพของโครงงานให้สูงขึ้น เช่น การใช้ IR Sensor ติดตั้งที่ความสูงต่างกันจะสามารถคาดคะเนได้ว่าสิ่งที่ตรวจจับได้เป็นสัตว์หรือมนุษย์

5. เพื่อให้ผลการทดสอบดูมีความน่าเชื่อถือมากขึ้นควรทำการทดสอบรูปคลื่นสัญญาณของเซนเซอร์เมื่อตรวจจับสิ่งมีชีวิตได้แล้วนำไปเทียบกับกราฟอ้างอิง

บรรณานุกรม

1) ปิยะชัย ควรปราโมทย์. (2553). **เซ็นเซอร์ตรวจจับความเคลื่อนไหว (Motion Sensor).** สืบค้น 10 กันยายน 2565, จาก http://application-with-embedded-linux.blogspot.com/2010/12/motion-sensor.html

2) มาโนชญ์ แสงศิริ. (2562). **Blynk : IoT Platform สนับสนุนจินตนาการสำหรับนวัตกร**. สืบค้น 10 ตุลาคม 2565, จาก https://www.scimath.org/article-technology/item/9820-blynk-iot-platform

3) ไอแซก. (2564). **ESP32-CAM: สิ่งที่คุณควรรู้เกี่ยวกับโมดูลนี้.** สืบค้น 10 ตุลาคม 2565, จาก https://www.hwlibre.com/th/esp32-%E0%B8%9A%E0%B9%81%E0%B8%84%E0%B8%A1/

4) ไม่ปรากฏชื่อผู้เขียน. (2563). **กลอนไฟฟ้า Solenoid Electromagnetic Door Lock LY-03 12V/350mA.** สืบค้น 10 กันยายน 2565, จาก https://www.arduino4.com/product/793

5) ไม่ปรากฏชื่อผู้เขียน. (2560). **Linear Solenoid Actuator**. สืบค้น 10 กันยายน 2565, จาก https://www.electronics-tutorials.ws/io/io_6.html

6) ไม่ปรากฏชื่อผู้เขียน. **IoT คืออะไร.** สืบค้น 10 ตุลาคม 2565, จาก https://aws.amazon.com/th/what-is/iot/

7) Chanida Phutthiwachananon. (2562). **วิธีเชื่อมต่อ LINE Notify แจ้งเตือนอัตโนมัติ ฉบับมือใหม่ เข้าใจง่าย ไม่ต้องเขียนโค้ด.** สืบค้น 10 ตุลาคม 2565, จาก https://digitalarea.co/line-notify-beginner/

8) Noppadon B. (2565). **แบตเตอรี่ลิเธี่ยมมีกี่ชนิด (What are types of Lithium Battery ?).** สืบค้น 10 พฤษจิกายน 2565, จาก https://www.klcbright.com/lithium-battery-type.php

9) Zahid Ali. (2562). **Introduction to HC-SR501**. สืบค้น 10 กันยายน 2565, จาก https://www.theengineeringprojects.com/2019/01/introduction-to-hc-sr501.html

ภาคผนวก

HC-SR501 PIR MOTION DETECTOR

Product Discription

HC-SR501 is based on infrared technology, automatic control module, using Germany imported LHI778 probe design, high sensitivity, high reliability, ultra-low-voltage operating mode, widely used in various auto-sensing electrical equipment, especially for battery-powered automatic controlled products.

Specification:

- Voltage: 5V – 20V
- Power Consumption: 65mA
- TTL output: 3.3V, 0V
- Delay time: Adjustable (.3->5min)
- Lock time: 0.2 sec
- Trigger methods: L – disable repeat trigger, H enable repeat trigger
- Sensing range: less than 120 degree, within 7 meters
- Temperature: – 15 ~ +70
- Dimension: 32*24 mm, distance between screw 28mm, M2, Lens dimension in diameter: 23mm

Application:

Automatically sensing light for Floor, bathroom, basement, porch, warehouse, Garage, etc, ventilator, alarm, etc.

Features:

- Automatic induction: to enter the sensing range of the output is high, the person leaves the sensing range of the automatic delay off high, output low.
- Photosensitive control (optional, not factory-set) can be set photosensitive control, day or light intensity without induction.
- Temperature compensation (optional, factory reset): In the summer when the ambient temperature rises to 30 ° C to 32 ° C, the detection distance is slightly shorter, temperature compensation can be used for performance compensation.
- Triggered in two ways: (jumper selectable)
  - non-repeatable trigger: the sensor output high, the delay time is over, the output is automatically changed from high level to low level;
  - repeatable trigger: the sensor output high, the delay period, if there is human activity in its sensing range, the output will always remain high until the people left after the delay will be high level goes low (sensor module detects a time delay period will be automatically extended every human activity, and the starting point for the delay time to the last event of the time).
- With induction blocking time (the default setting: 2.5s blocked time): sensor module after each sensor output (high into low), followed by a blockade set period of time, during this time period sensor does not accept any sensor signal. This feature can be achieved sensor output time "and" blocking time "interval between the work can be applied to interval detection products; This function can inhibit a variety of interference in the process of load switching. (This time can be set at zero seconds – a few tens of seconds).
- Wide operating voltage range: default voltage DC4.5V-20V.
- Micropower consumption: static current <50 microamps, particularly suitable for battery-powered automatic control products.
- Output high signal: easy to achieve docking with the various types of circuit.

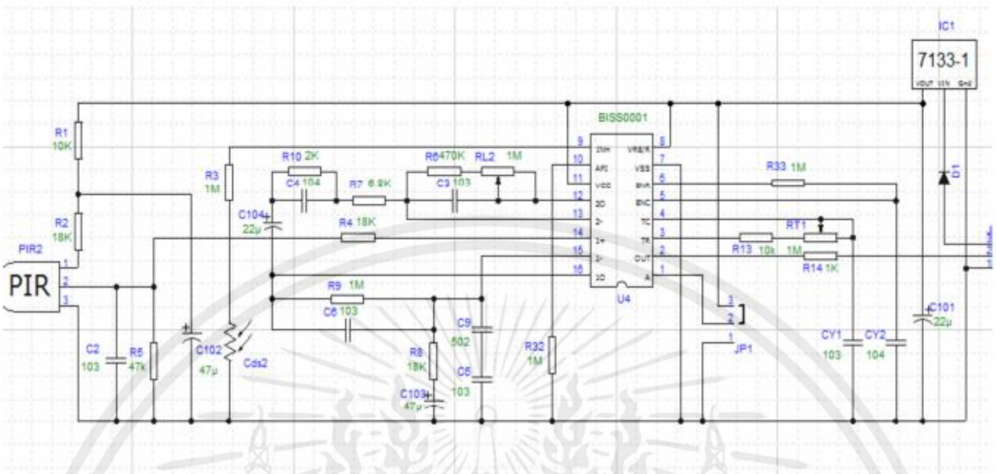
Adjustment:

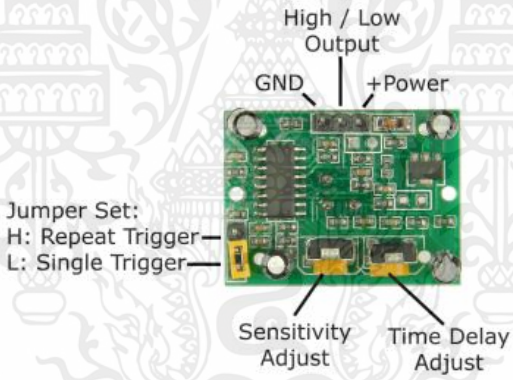
- Adjust the distance potentiometer clockwise rotation, increased sensing distance (about 7 meters), on the contrary, the sensing distance decreases (about 3 meters).
- Adjust the delay potentiometer clockwise rotation sensor the delay lengthened (300S), on the contrary, shorten the induction delay (5S).

Instructions for use:

- Sensor module is powered up after a minute, in this initialization time intervals during this module will output 0-3 times, a minute later enters the standby state.
- Should try to avoid the lights and other sources of interference close direct module surface of the lens, in order to avoid the introduction of interference signal malfunction; environment should avoid the wind flow, the wind will cause interference on the sensor.
- Sensor module with dual probe, the probe window is rectangular, dual (A B) in both ends of the longitudinal direction
  - so when the human body from left to right or right to left through the infrared spectrum to reach dual time, distance difference, the greater the difference, the more sensitive the sensor,
  - when the human body from the front to the probe or from top to bottom or from bottom to top on the direction traveled, double detects changes in the distance of less than infrared spectroscopy, no difference value the sensor insensitive or does not work;
- The dual direction of sensor should be installed parallel as far as possible in inline with human movement. In order to increase the sensor angle range, the module using a circular lens also makes the probe surrounded induction, but the left and right sides still up and down in both directions sensing range, sensitivity, still need to try to install the above requirements.

HC-SR501 PIR MOTION DETECTOR





1 working voltage range :DC 4.5-20V

2 Quiescent Current :50uA

3 high output level 3.3 V / Low 0V

4. Trigger L trigger can not be repeated / H repeated trigger

5. circuit board dimensions :32 * 24 mm

6. maximum 110 ° angle sensor

7. 7 m maximum sensing distance

Product Type	HC--SR501 Body Sensor Module
Operating Voltage Range	5-20VDC
Quiescent Current	<50uA
Level output	High 3.3 V /Low 0V
Trigger	L can not be repeated trigger/H can be repeated trigger(Default repeated trigger)
Delay time	5-300S(adjustable) Range (approximately .3Sec -5Min)
Block time	2.5S(default)Can be made a range(0.xx to tens of seconds
Board Dimensions	32mm*24mm
Angle Sensor	<110 ° cone angle
Operation Temp.	-15-+70 degrees
Lens size sensor	Diameter:23mm(Default)

Application scope
•Security products
•Body induction toys
•Body induction lamps
•Industrial automation control etc

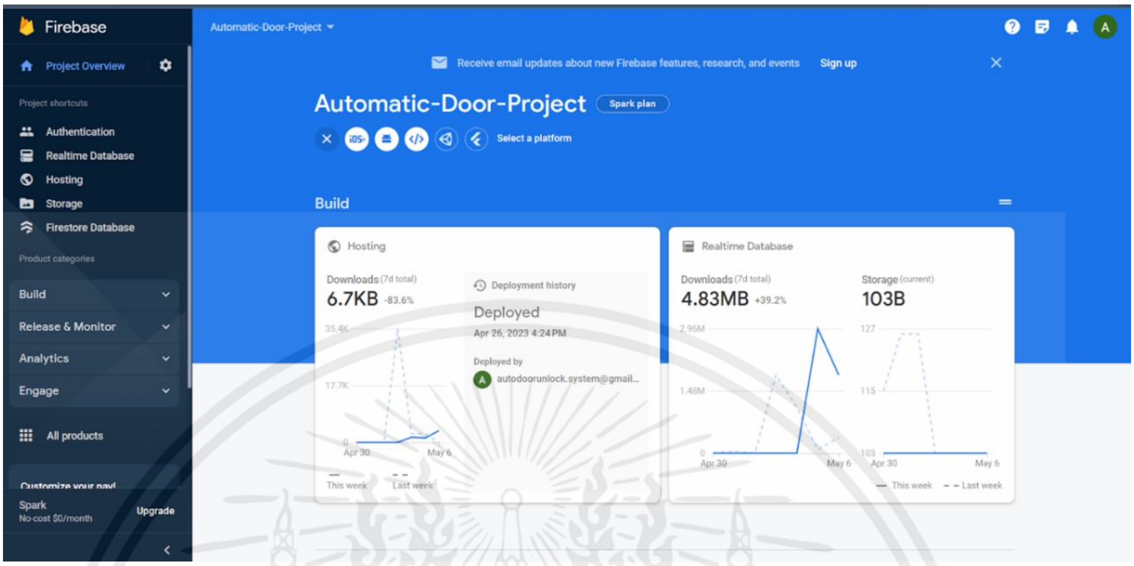
Pyroelectric infrared switch is a passive infrared switch which consists of BISS0001 ,pyroelectric infrared sensors and a few external components. It can au open all kinds of equipments, inculding incandescent lamp, fluorescent lamp, intercom, automatic, electric fan, dryer and automatic washing machine, etc. It is widely used in enterprises, hotels, stores, and corridor and other sensitive area for automatical lamplight, lighting and alarm system.

Instructions

Induction module needs a minute or so to initialize. During initializing time, it will output 0-3 times. One minute later it comes into standby.
Keep the surface of the lens from close lighting source and wind, which will introduce interference.
Induction module has double -probe whose window is rectangle. The two sub-probe (A and B) is located at the two ends of rectangle. When human body r to right, or from right to left, Time for IR to reach to reach the two sub-probes differs.The lager the time diffference is, the more sensitive this module is. Wh body moves face-to probe, or up to down, or down to up, there is no time difference. So it does not work. So instal the module in the direction in which mos activities behaves, to guarantee the induction of human by dual sub-probes. In order to increase the induction range, this module uses round lens which ca from all direction. However, induction from right or left is more sensitivity than from up or down.

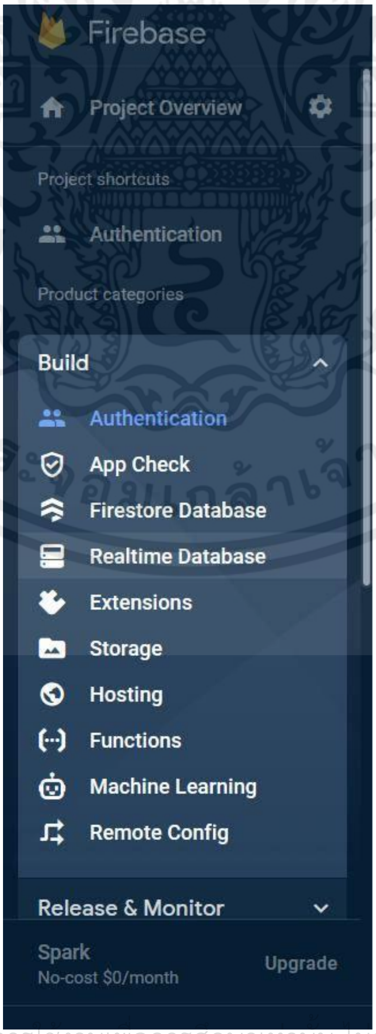
ขั้นตอนการออกแบบ Firebase Realtime Database

- เปิดเว็บบราวเซอร์ไปที่ firebase.google.com ไปที่ Go to console
- จะปรากฏหน้าต่างโปรเจคใน Firebase



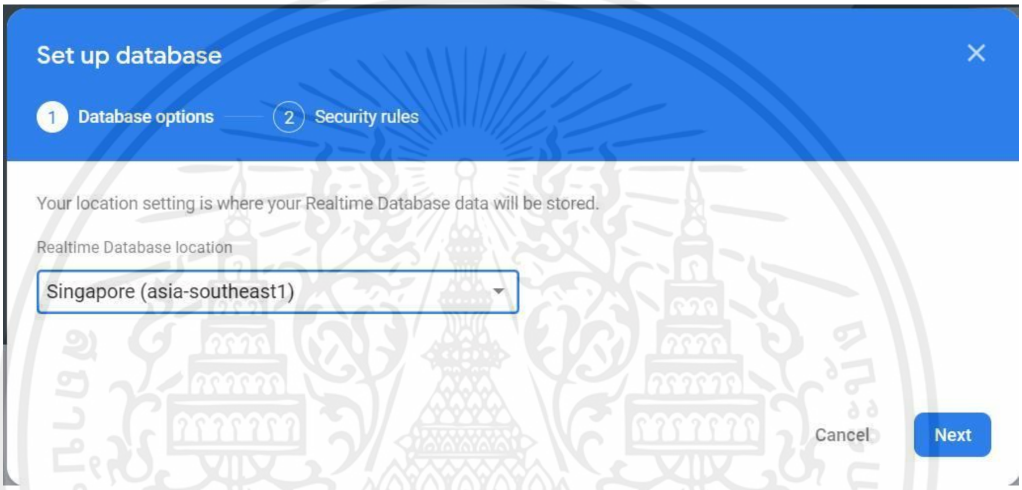
รูป Firebase Project Overview

- จากนั้นไปที่ Build > Realtime Database

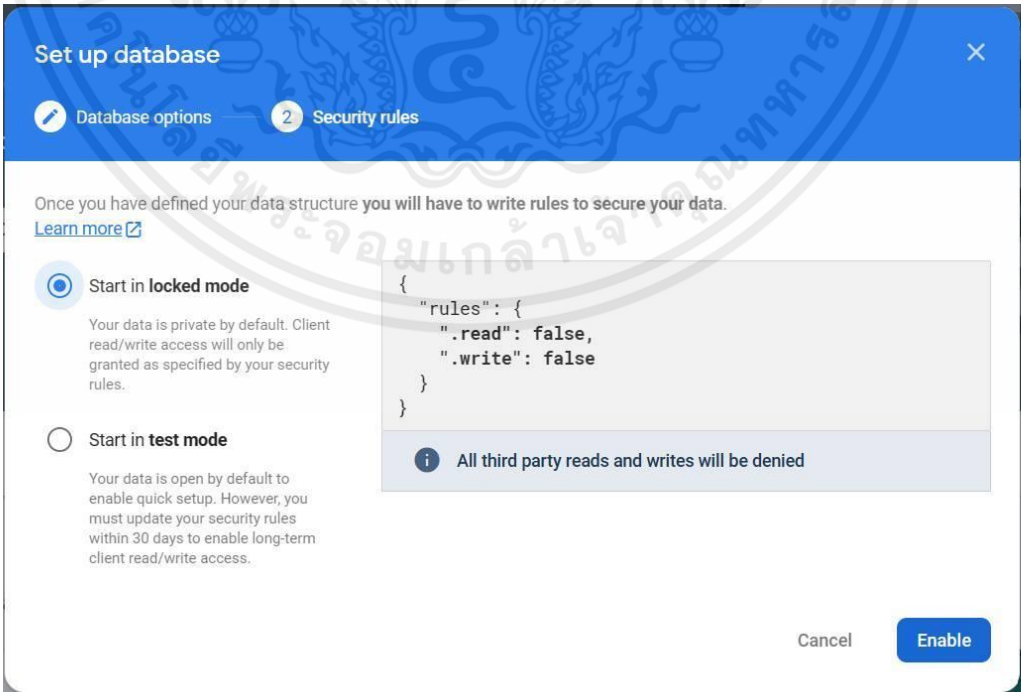


- คลิก Create Database

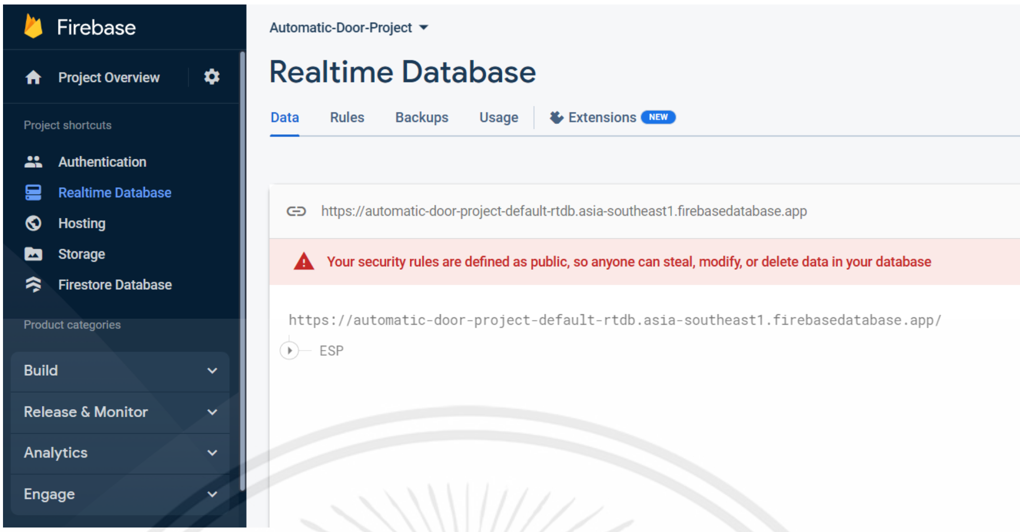
- เลือก Realtime Database location บริเวณที่ใกล้เคียงกับตำแหน่งปัจจุบันที่สุด แล้วกด Next



- เลือก Start in Test Mode แล้วกด Next



- จากนั้นจะปรากฏหน้าต่าง Realtime Database ดังรูป

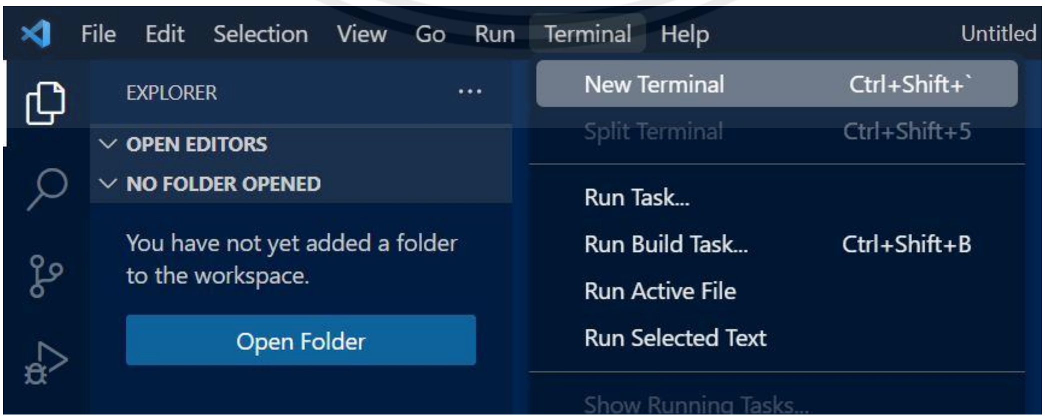


โดยการแสดงข้อมูลและควบคุมอุปกรณ์ที่เชื่อมกับบอร์ดไมโครคอนโทรลเลอร์ จะถูกส่งผ่าน Path ซึ่งสามารถสร้างขึ้นเองได้โดยการเขียนโค้ดใน Arduino IDE ได้แก่

- การสร้างข้อมูลแสดงสถานะของ PIR Sensor ใช้คำสั่ง Firebase.RTDB.setInt(&fbdo, "ESP/Device/Sensor/pir_Data", pirData)
- การสร้างข้อมูลแสดงหมายเลขโทรศัพท์จาก Keypad ใช้คำสั่ง Firebase.RTDB.setString(&fbdo, "ESP/Device/Keypad/keypad_Data", phoneinput)
- การสร้างข้อมูลแสดงสถานะของกลอนประตูไฟฟ้าและการควบคุมผ่านเว็บแอพ จะต้องเรียกการใช้งาน Object Firebasejson ก่อน จากนั้นจึงมีการสตรีมข้อมูลผ่านคำสั่ง Firebase.RTDB.beginStream(&stream, listenerPath.c_str()) โดยที่ listenerPath = "ESP/Device/Relaydoor/"

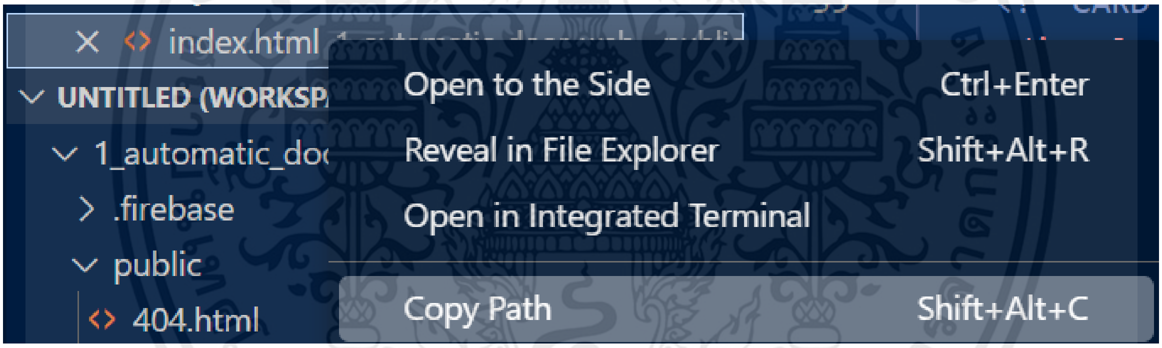
ขั้นตอนการทำ Firebase Web Hosting

- เปิด VS Code ไปที่ Terminal > New Terminal

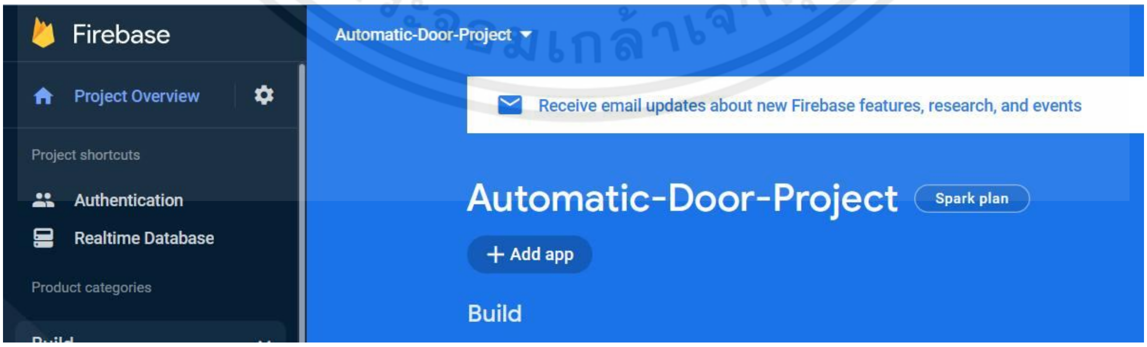


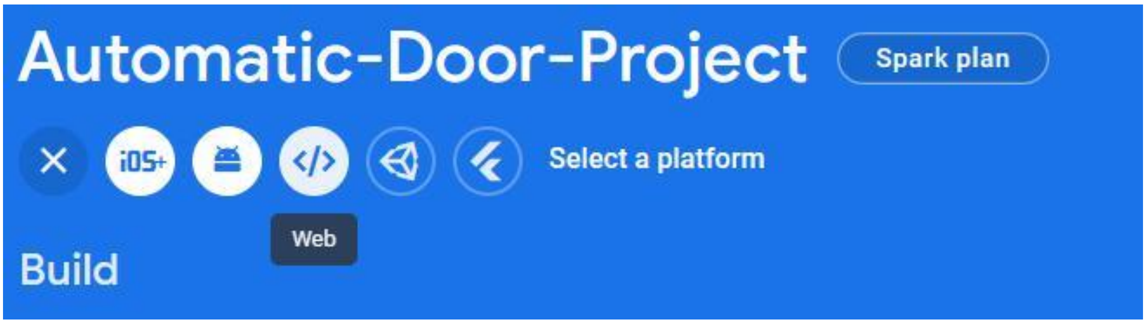
- พิมพ์คำสั่ง cd \ ต่อท้าย path ที่เคย install ไว้ จากนั้นกด enter , พิมพ์คำสั่ง npm -g install firebase-tools ต่อท้าย path ที่เคย install ไว้ จากนั้นกด enter, จะมีข้อความปรากฏเช่นนี้เมื่อการติดตั้ง Firebase Tools สำเร็จ

- จากนั้นให้ทำการสร้างโฟลเดอร์ สำหรับรวบรวมไฟล์ index.html style.css และ app.js ซึ่งสามารถคำสั่งสำหรับตกแต่งเว็บและเพิ่มฟังก์ชันการทำงานเชื่อมกับ Firebase ได้ตามใจชอบ หากต้องการดูหน้าเว็บแอพที่เขียนไว้ว่ามีหน้าตาอย่างไร ทำได้โดยกดปุ่ม Ctrl+Shift+E แล้วคลิกที่ไฟล์ index.html จากนั้นคลิกขวา ไปที่ Copy Path แล้วให้วาง Copy Path บนเว็บบราวเซอร์ได้เลย



- การเพิ่มเว็บแอพลิเคชันใน Firebase เปิด Console ของ Firebase ไปที่ Project Overview > + Add app แล้วเลือก Web app icon





- เพิ่ม Firebase SDK (Software Development Kit) โดยใช้ npm (Node Package Manager) ซึ่งเป็นคือเครื่องมือในการจัดการกับ package ของ Node.js โปรเจค ซึ่งถูกเขียนด้วยภาษา JavaScript

- กลับไปที่ VS Code ไปที่ Terminal > New Terminal ใช้คำสั่ง firebase login หากสำเร็จจะปรากฏข้อความเช่นนี้

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

Waiting for authentication...

+ Success! Logged in as autodoorunlock.system@gmail.com
PS C:\>
```

- ต่อมาเป็นการติดตั้ง Firebase Hosting ไปที่ Terminal ใช้คำสั่ง firebase init แล้วกด Enter จากนั้นเลือก Yes แล้วเลือก features directory ดังต่อไปนี้

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL                                    node
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices.
 (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
 (*) Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance
 ( ) Firestore: Configure security rules and indexes files for Firestore
 ( ) Functions: Configure a Cloud Functions directory and its files
>(*) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
 ( ) Hosting: Set up GitHub Action deploys
 (*) Storage: Configure a security rules file for Cloud Storage
 ( ) Emulators: Set up local emulators for Firebase products
(Move up and down to reveal more choices)
```

- จากนั้นเลือกที่ Use an existing project แล้วกด enter

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option:
> Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project
```

- เลือก automatic-door-project

```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: Use an existing project
? Select a default Firebase project for this directory:
> automatic-door-project (Automatic-Door-Project)
  fir-cam-2d873 (Demo Cam)
```

- เมื่อทำการเขียนเว็บแอพผ่าน VS Code เสร็จแล้ว ให้ทำการ Deploy Website ขึ้น Web Hosting ของ Firebase ด้วยคำสั่ง firebase deploy ผ่านหน้าจอ Terminal แล้วกด enter

Source Code สำหรับการทำงาน

- ESP32-CAM

```
#include <WiFi.h>
#include <WiFiClient.h>
#include "esp_camera.h"
#include "esp_system.h"
#include "Arduino.h"
#include "soc/soc.h"           // Disable brownout problems
#include "soc/rtc_cntl_reg.h"  // Disable brownout problems

#include <TridentTD_LineNotify.h>
#define LINE_TOKEN  "V5PODWLLYM8jNmdsO4zPC2jTzeXMfufiVC3DkGTkDMf"
#define WIFI_SSID "OPPO A9 2020_E"   //WiFi name
#define WIFI_PASSWORD "tk8rsumb"   //PASSWORD
//#define WIFI_SSID "fakfahwifi"
//#define WIFI_PASSWORD "payme2000"
//#define APP_DEBUG

// Pin definition for CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
```

```
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
// Define GPIO pins
#define PIR_SENSOR 13
int PIRState = 0;

// Initialize WiFi
void initWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.println('.');
    delay(500);
  }
  Serial.println(WiFi.localIP());
  Serial.println();
}

void setup() {
  Serial.begin(115200);
  //setup pinMode of each modules
  pinMode(PIR_SENSOR, INPUT);

//เรียกการใช้ ISR จาก PIR sensor
  initWiFi();
  LINE.setToken(LINE_TOKEN);
  esp_sleep_enable_ext0_wakeup((gpio_num_t)GPIO_NUM_13, HIGH);
 //Define the camera settings
  camera_config_t configs;
```

```
  configs.ledc_channel = LEDC_CHANNEL_0;
  configs.ledc_timer = LEDC_TIMER_0;
  configs.pin_d0 = Y2_GPIO_NUM;
  configs.pin_d1 = Y3_GPIO_NUM;
  configs.pin_d2 = Y4_GPIO_NUM;
  configs.pin_d3 = Y5_GPIO_NUM;
  configs.pin_d4 = Y6_GPIO_NUM;
  configs.pin_d5 = Y7_GPIO_NUM;
  configs.pin_d6 = Y8_GPIO_NUM;
  configs.pin_d7 = Y9_GPIO_NUM;
  configs.pin_xclk = XCLK_GPIO_NUM;
  configs.pin_pclk = PCLK_GPIO_NUM;
  configs.pin_vsync = VSYNC_GPIO_NUM;
  configs.pin_href = HREF_GPIO_NUM;
  configs.pin_sscb_sda = SIOD_GPIO_NUM;
  configs.pin_sscb_scl = SIOC_GPIO_NUM;
  configs.pin_pwdn = PWDN_GPIO_NUM;
  configs.pin_reset = RESET_GPIO_NUM;
  configs.xclk_freq_hz = 20000000;
  configs.pixel_format = PIXFORMAT_JPEG;
  //settings for a camera with PSRAM
  if(psramFound()){
    configs.frame_size = FRAMESIZE_UXGA; // FRAMESIZE_ +
QVGA|CIF|VGA|SVGA|XGA|SXGA|UXGA
    configs.jpeg_quality = 10;
    configs.fb_count = 2;
  } else {
    configs.frame_size = FRAMESIZE_SVGA;
    configs.jpeg_quality = 12;
    configs.fb_count = 1;
  }
  // Initialize Camera
  esp_err_t err = esp_camera_init(&configs);
```

```
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }
} //End of Setup function

void Camera_capture() {
  camera_fb_t * fb = NULL;
  delay(200);
  // Take Picture with Camera
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    return;
  }
   Send_line(fb->buf,fb->len);
   esp_camera_fb_return(fb);
}
void Send_line(uint8_t *image_data,size_t   image_size){
   LINE.notifyPicture("Motion Detected!",image_data, image_size);
   Serial.println("Photo sent to line");
  }
void loop() {
  PIRState = digitalRead(PIR_SENSOR);
  Serial.print("val = ");
  Serial.println(PIRState);
  if (PIRState == 1) {
    if (millis() > 5 * 1000) { // ถ้าครบ 5 วินาที
    Camera_capture();
    }
   } else {
    delay(100);
    Serial.println("Going to sleep now");
```

```
    esp_deep_sleep_start(); // เข้าสู่โหมด Deep Sleep
    }
} //End of loop function
```

- ESP32

```
#include <Arduino.h>
#include <WiFi.h>
#include <Firebase_ESP_Client.h>
#include "addons/TokenHelper.h"
#include "addons/RTDBHelper.h"

#define WIFI_SSID "OPPO A9 2020_E"
#define WIFI_PASSWORD "tk8rsumb"

#define FIREBASE_HOST "automatic-door-project-default-rtdb.asia-
southeast1.firebasedatabase.app"
#define FIREBASE_AUTH "lS9OC9zVDlS92VSqt1FxrgdHJ5kDNBT1EPMG5i7Q"

#define API_KEY "AIzaSyAhfMv_arzn3TRHlhcMA18D7DSizgnuXvw"
#define DATABASE_URL "https://automatic-door-project-default-rtdb.asia-
southeast1.firebasedatabase.app/"
#include <Keypad_I2C.h>
#include <Keypad.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
#define I2CADDR 0x20  // กำหนด Address ของ I2C

// Define GPIO pins
#define PIR_SENSOR 13
#define RELAY_DOOR 16
#define SDA 21 /*I2C Pin for ESP32*/
#define SCL 22 /*SDA เขียว, SCL เหลือง*/
```

```
// Create Firebase objects
FirebaseData stream;
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
String listenerPath = "ESP/Device/Relaydoor/";
const byte ROWS = 4;  // กำหนดจำนวนของ Rows
const byte COLS = 4;  // กำหนดจำนวนของ Columns
// กำหนด Key ที่ใช้งาน (4x4)
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9','C'},
  {'*','0','#'}
};
// กำหนด Pin ที่ใช้งาน (4x4)
byte rowPins[ROWS] = {0, 1, 2, 3}; // เชื่อมต่อกับ Pin แถวของปุ่มกด
byte colPins[COLS] = {4, 5, 6, 7}; // เชื่อมต่อกับ Pin คอลัมน์ของปุ่มกด
Keypad_I2C keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS, I2CADDR, PCF8574 );

// Create variable
char key;
String phoneinput = "";
byte phonedigit = 0;
boolean signupOK = false;
int pirData = 0;

// Initialize WiFi
void initWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.println('.');
```

```
    delay(500);
  }
  Serial.println(WiFi.localIP());
  Serial.println();
}
// Callback function that runs on database changes
void streamCallback(FirebaseStream data){
  Serial.printf("stream path, %s\nevent path, %s\ndata type, %s\nevent type, %s\n\n",
              data.streamPath().c_str(),
              data.dataPath().c_str(),
              data.dataType().c_str(),
              data.eventType().c_str());
  printResult(data);
  Serial.println();
  // Get the path that triggered the function
  String streamPath = String(data.dataPath());
  // if the data returned is an integer, there was a change on the GPIO state on the following
path /{gpio_number}
  if (data.dataTypeEnum() == fb_esp_rtdb_data_type_integer){
    String gpio = streamPath.substring(1);
    int state = data.intData();
    Serial.print("GPIO: ");
    Serial.println(gpio);
    Serial.print("STATE: ");
    Serial.println(state);
    digitalWrite(gpio.toInt(), state);
  }
  /* When it first runs, it is triggered on the root (/) path and returns a JSON with all keys
  and values of that path. So, we can get all values from the database and updated the
GPIO states*/
  if (data.dataTypeEnum() == fb_esp_rtdb_data_type_json){
    FirebaseJson json = data.to<FirebaseJson>();
  // To iterate all values in Json object
```

```
    size_t count = json.iteratorBegin();
    Serial.println("\n---------");
    for (size_t i = 0; i < count; i++){
        FirebaseJson::IteratorValue value = json.valueAt(i);
        int gpio = value.key.toInt();
        int state = value.value.toInt();
        Serial.print("STATE: ");
        Serial.println(state);
        Serial.print("GPIO:");
        Serial.println(gpio);
        digitalWrite(gpio, state);
        Serial.printf("Name: %s, Value: %s, Type: %s\n", value.key.c_str(), value.value.c_str(),
value.type == FirebaseJson::JSON_OBJECT ? "object" : "array");
    }
    Serial.println();
    json.iteratorEnd(); // required for free the used memory in iteration (node data collection)
  }
  Serial.printf("Received stream payload size: %d (Max. %d)\n\n", data.payloadLength(),
data.maxPayloadLength());
}

void streamTimeoutCallback(bool timeout){
  if (timeout)
    Serial.println("stream timeout, resuming...\n");
  if (!stream.httpConnected())
    Serial.printf("error code: %d, reason: %s\n\n", stream.httpCode(),
stream.errorReason().c_str());
}

void Hello_Welcome() {
  lcd.display();
  lcd.backlight();
  lcd.clear();
```

```
  lcd.setCursor(0, 0);
  lcd.print("  Hello, Welcome!  ");
  lcd.setCursor(0, 1);
  lcd.print("Enter phone number : ");
} //End of Hello_Welcome function

void sentPhone(){
   //todo send to firebase or line noti
   Serial.println("Phone = "+phoneinput);
   if(Firebase.RTDB.setString(&fbdo, "ESP/Device/Keypad/keypad_Data", phoneinput)) {
   // if sentInt if success it will print keypad data
    Serial.println();
    Serial.print(phoneinput);
    Serial.println("Pass");
    Serial.println("Path: " + fbdo.dataPath());
    Serial.println("Type: " + fbdo.dataType());
   } else {
    Serial.println("Failed");
    Serial.println("REASON: " + fbdo.errorReason());
   } // End of Send keypad data
        phoneinput = "";
        phonedigit = 0;
  }

void keypadpressed() {
if(key) {
 if(key == 'C') {
    lcd.clear();
    Hello_Welcome();
    //display message to visitor
}
 else if(key) {
```

```
  if (key == '1' || key == '2' || key == '3' || key == '4' || key == '5' || key == '6' || key == '7' || key
== '8' || key == '9' || key == '0') {
    lcd.clear();
    Hello_Welcome();
    //แสดงข้อความต้อนรับ
    phoneinput += key;
    phonedigit++;
    Serial.println(phoneinput);
    lcd.setCursor(0, 2);
    lcd.print(String(phoneinput));
}
 if (key == '*') {
    lcd.clear();
    Hello_Welcome();
    lcd.setCursor(0, 2);
    lcd.print("Deleted");
    Serial.println("Deleted");
    phoneinput = "";
    phonedigit = 0;
    lcd.setCursor(0, 3);
    lcd.print("                    ");
    delay(1000);
   // lcd.clear();
}

 if (key == '#') {
    if (phonedigit == 10) {
        lcd.clear();
        Hello_Welcome();
        lcd.setCursor(0, 2);
        lcd.print("        OK        ");
        Serial.println("        OK        ");
        //phoneinput = "";
```

```
        //phonedigit = 0;
        sentPhone();
        delay(1000);
        lcd.clear();
        } else {
  //      if (phonedigit < 10 || phonedigit > 10) {
        phoneinput = "";
        phonedigit = 0;
        //lcd.clear();
        Hello_Welcome();
        lcd.setCursor(0, 2);
        lcd.print("Not Correct");
        lcd.setCursor(0, 3);
        lcd.print("                    ");
        delay(1000);
        //lcd.clear();
         }
      } //end of '#' (Enter)
    } //end of get phone no.
  } //end of key-pressed check
} //end of keypadpressed function

void setup() {
  Serial.begin(115200);
  initWiFi();
  // Initialize Keypad
  Wire.begin(SDA, SCL);
  keypad.begin(makeKeymap(keys));  // เรียกการเชื่อมต่อ Keypad
  // Initialize LCD
  lcd.init();
  lcd.backlight();
  lcd.clear();
  // Initialize Outputs
```

```
  pinMode(PIR_SENSOR, INPUT_PULLUP);
  pinMode(RELAY_DOOR,OUTPUT);
  config.api_key = API_KEY;
  config.database_url = DATABASE_URL;

    if (Firebase.signUp(&config, &auth, "", "")){
    Serial.println("ok");
    signupOK = true;
  }
  else{
    Serial.printf("%s\n", config.signer.signupError.message.c_str());
  }
  Firebase.reconnectWiFi(true);
  config.token_status_callback = tokenStatusCallback;
  // Assign the maximum retry of token generation
  config.max_token_generation_retry = 5;
  Firebase.begin(&config, &auth);
  // Streaming (whenever data changes on a path)
  // Begin stream on a database path --> board1/outputs/digital
  if (!Firebase.RTDB.beginStream(&stream, listenerPath.c_str()))
    Serial.printf("stream begin error, %s\n\n", stream.errorReason().c_str());
  // Assign a calback function to run when it detects changes on the database
  Firebase.RTDB.setStreamCallback(&stream, streamCallback, streamTimeoutCallback);
  delay(200);
//เรียกการใช้ Sleepmode
esp_sleep_enable_ext0_wakeup((gpio_num_t)GPIO_NUM_13, HIGH);
} //End of Setup function

void loop() {
  if (Firebase.ready() && signupOK) {
      // Store pir data to a rtdb (setInt)
    pirData = digitalRead(PIR_SENSOR);
  key = keypad.getKey(); //เก็บค่าที่ได้จากการกด Keypad
```

```
  keypadpressed();

   if(Firebase.RTDB.setInt(&fbdo, "ESP/Device/Sensor/pir_Data", pirData)) {
   // if sentInt if success it will print pir data
    Serial.println();
    Serial.print("PIR State = ");
    Serial.print(pirData);
    Serial.println("Pass");
    Serial.println("Path: " + fbdo.dataPath());
    Serial.println("Type: " + fbdo.dataType());
   } else {
    Serial.println("Failed");
    Serial.println("REASON: " + fbdo.errorReason());
   } // End of Send pir data

    if(pirData == 0) {
    if (millis() > 3 * 1000) { // ถ้าครบ 3 วินาที
      lcd.clear();
      lcd.print("System Off");
      delay(1000); // หน่วงเวลา 1 วินาทีก่อนที่ lcd จะดับและ esp เข้า sleep mode
      lcd.noDisplay();
      lcd.noBacklight();
      lcd.clear(); //system off and lcd will not display until press 'C'
    Serial.println("Going to sleep now");
    esp_deep_sleep_start(); // เข้าสู่โหมด Deep Sleep
    }
  }
    } //end of firebase ready check
} //End of loop function
```